②

ARMY RESEARCH LABORATORY

A Guide to FASTGEN Target Geometric Modeling

Edward D. Aitken
Susan L. Jones
Allen W. Dean

ARL-CR-69 October 1993

prepared by

ASI Systems International
Eglin Group
838 N. Eglin Parkway, Suite 421
Fort Walton Beach, Florida 32548

under contract

DAAA15-88-D-0019

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION IS UNLIMITED.

NOTICES

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	October 1993	Final, 1 Jun 92 - 1 Jun 93

4. TITLE AND SUBTITLE

A Guide to FASTGEN Target Geometric Modeling

5. FUNDING NUMBERS

C: DAAA15-88-D-0019

6. AUTHOR(S)

Edward D. Aitken, Susan L. Jones, and Allen W. Dean

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

ASI Systems International
Eglin Group
838 N. Elgin Parkway, Suite 421
Fort Walton Beach, FL 32548

8. PERFORMING ORGANIZATION REPORT NUMBER

JTCG/AS-92-SM-025

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

U.S. Army Research Laboratory
ATTN: AMSRL-OP-CI-B (Tech Lib)
Aberdeen Proving Ground, MD 21005-5066

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

ARL-CR-69

11. SUPPLEMENTARY NOTES

The Point of Contact for this report is Dr. Abdul R. Kiwan, U.S. Army Research Laboratory, ATTN: AMSRL-SL-BA, Aberdeen Proving Ground, MD 21005-5068.

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

Vulnerability analysis programs, such as COVART, require target description data of sufficient detail and completeness to represent the physical and geometric detail of a target model from any attack aspect. The FASTGEN 3 computer model is used to generate the target description by developing a listing of the physical dimensions of target components, component location, and air spaces encountered along parallel shotlines passing through the target from a specified attack direction.

A prerequisite to the execution of FASTGEN is the development of a geometric description of a target whose exterior and interior components surfaces are described using triangles, spheres, cones, cylinders, and rods. The objective of this manual is to provide a guide to experienced and inexperienced model developers to assist them in developing geometric models using the CONVERT computer code to generate target description data in the FASTGEN format. Specific model preparation procedures, recommended procedures, frequently encountered pitfalls and proven shortcut model preparation procedures are discussed. Emphasis is placed on the use of CONVERT constructs in model development

14. SUBJECT TERMS	15. NUMBER OF PAGES
vulnerability; target recognition, geometric target description; geometric modeling	118
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

INTENTIONALLY LEFT BLANK.

PREFACE

This technical manual documents efforts performed by ASI Systems International, 838 North Eglin Parkway, Fort Walton Beach, Florida, 32547, under Contract DAAA15-88-D-0019, dated April 1993. The government Project Manager was Mr. Daniel McInnis, Aeronautical Systems Center (ASC/XREWS), Eglin Air Force Base, Florida, 32542-5434. The Program Monitor was Mr. B. Tate Bentley, Army Research Laboratory (ARL), Aberdeen Proving Ground, MD, 21005-5071. The principal investigator at ASI was Mr. Edward D. Aitken with assistance from Ms. Susan L. Jones and Allen W. Dean.

Vulnerability programs such as PDAM and COVART require target line-of-sight (LOS) description inputs of sufficient detail and completeness to represent the target ballistically from any attack aspect. FASTGEN provides this description by developing listings of target components and air spaces encountered along a large number of uniformly distributed parallel rays emanating from a specific direction and passing through the target.

A prerequisite to FASTGEN execution is the preparation of a geometric model where exterior and interior component surfaces are described in terms of triangles, spheres, cones, cylinders, and rods. The purpose of this manual is to provide a guide to assist analysts in the creation of target geometric models (TGM) using the CONVERT computer program to generate target description data in FASTGEN format. Specific model-preparation procedures are discussed as are frequently encountered pitfalls and proven shortcut model-preparation methods. Emphasis is placed on use of CONVERT constructs in model development.

INTENTIONALLY LEFT BLANK.

TABLE OF CONTENTS

TABLE OF CONTENTS (CONCLUDED)

LIST OF FIGURES

LIST OF FIGURES (CONTINUED)

LIST OF FIGURES (CONTINUED)

LIST OF FIGURES (CONCLUDED)

LIST OF TABLES

INTENTIONALLY LEFT BLANK.

SECTION I

INTRODUCTION

This document presents an evolutionary treatise of target modeling techniques and discusses the purpose, level of detail, and modeling approach used in defining the target geometric model (TGM). FASTGEN modeling conventions, constructs, and definitions are presented as well as concepts and rationale governing the assignment of component code numbers and space codes, the application of volume mode and plate mode modeling techniques, the construction and use of component sketches, the mirroring of components, and the use of volume subtraction. Peculiar considerations related to the intended use of a TGM with the FASTGEN program are presented followed by some elementary modeling examples. Discussions are concluded with comments pertaining to the use of outputs from CONVERT, (Reference 1) FASTGEN, (References 2 and 3), and selected plot programs to aid in debugging the TGM after it has been created.

1. BACKGROUND

Predicting levels of damage to the critical components of a target, given a specific warhead/target encounter, is a continuing requirement of survivability/vulnerability analysts. Reasonably accurate predictions can be used to economically compare the effectiveness of different warheads against a known target or target array. Predictions also provide a means of specifying the need for improvements in both offensive and defensive weapon systems. Such a need might be in warhead design or in shielding by the application of new developments in material science (exotic composites, ceramics, radar/millimeter wave (MMW) absorbency, etc.). Perhaps the best known use of target damage predictions is in support of end-game effectiveness analysis where the quantity of munitions, the delivery platform, and tactics become players. The accepted practice of generating target damage predictions has traditionally involved a triad of disciplines defined as target modeling, warhead lethality modeling, and the development of component criticality in the form of a failure mode effects criticality analyses (FMECA). The first leg of this triad, target modeling, is probably the most difficult and demanding of the three.

There has been a great deal of time and effort expended over the years in attempting to best describe a target. Early techniques consisted of scaled drawings which were true oblique projections based on the attack aspect. An attack aspect is defined by an azimuth angle and

an elevation angle. As shown in Figure 1, the azimuth angle is measured in the horizontal plane of the target, positive in the counterclockwise direction. The elevation angle is then measured in the azimuthal plane previously established. The view or projection plane is thereby established as a plane normal to the viewing line. It is from the true oblique target projection upon this plane that areas of critical components could be determined by either measuring the areas projected by the components or by calculating them mathematically. Less accurate (but sometimes all that was available) were actual photographs of the target. Eventually, computer programs were developed to aid in the generation of target description data. This automated approach greatly increased flexibility to accommodate various attack aspects and changes to basic target descriptive data.

Several methods of target component surface fitting were attempted during the early stages of computer application. These included fitting of appropriate known regular surface loci as an approximation to a given surface, polynomial surface fitting, and surface area approximation by plane segment partitioning. The latter method was the form finally selected and triangles were the chosen primitive with which to approximate surface areas. This method allows any surface, flat or curved, exterior or interior, to be approximated by describing it as a series of one or more consecutively adjacent triangles whose points (vertices) are located in three-dimensional space. Flat surfaces can be described with large triangles and a few smaller ones if the surface is irregular. Curved surfaces can be described using several small triangles, with the size of the triangle decreasing if increased accuracy is desired. Figure 2 illustrates this concept. Note that any three consecutively sequenced points define a triangle.

During a period from late 1960 to mid 1970, several programs were written by Falcon Research and Development Company of Denver, Colorado, to assist in model preparation, to generate binary geometric target model files, and to produce a series of parallel line penetration descriptions of the target commonly called "shotline" descriptions.

The computer program which generated the shotlines was SHOTGEN (Reference 4). It required two input files for execution. One file contained the geometric target model where all surfaces were described using triangles except for wires, cables, control rods, hydraulic lines, fuel lines, etc., which could be described by an axis and a radius (called "influence" mode). The second file contained run specification cards which, among other things, defined the attack aspect(s) and whether or not the geometric target file was in binary-coded decimal (BCD) or binary form. Also input was the size of the grid squares to be superimposed over the target, normal to the attack aspect, and from which shotlines through the target would emanate as illustrated in Figure 3. SHOTGEN provided, as output, a detailed item-by-item listing of the components, surface thickness, air spaces, and entrance and exit obliquity angles encountered by the parallel rays passing through the target.

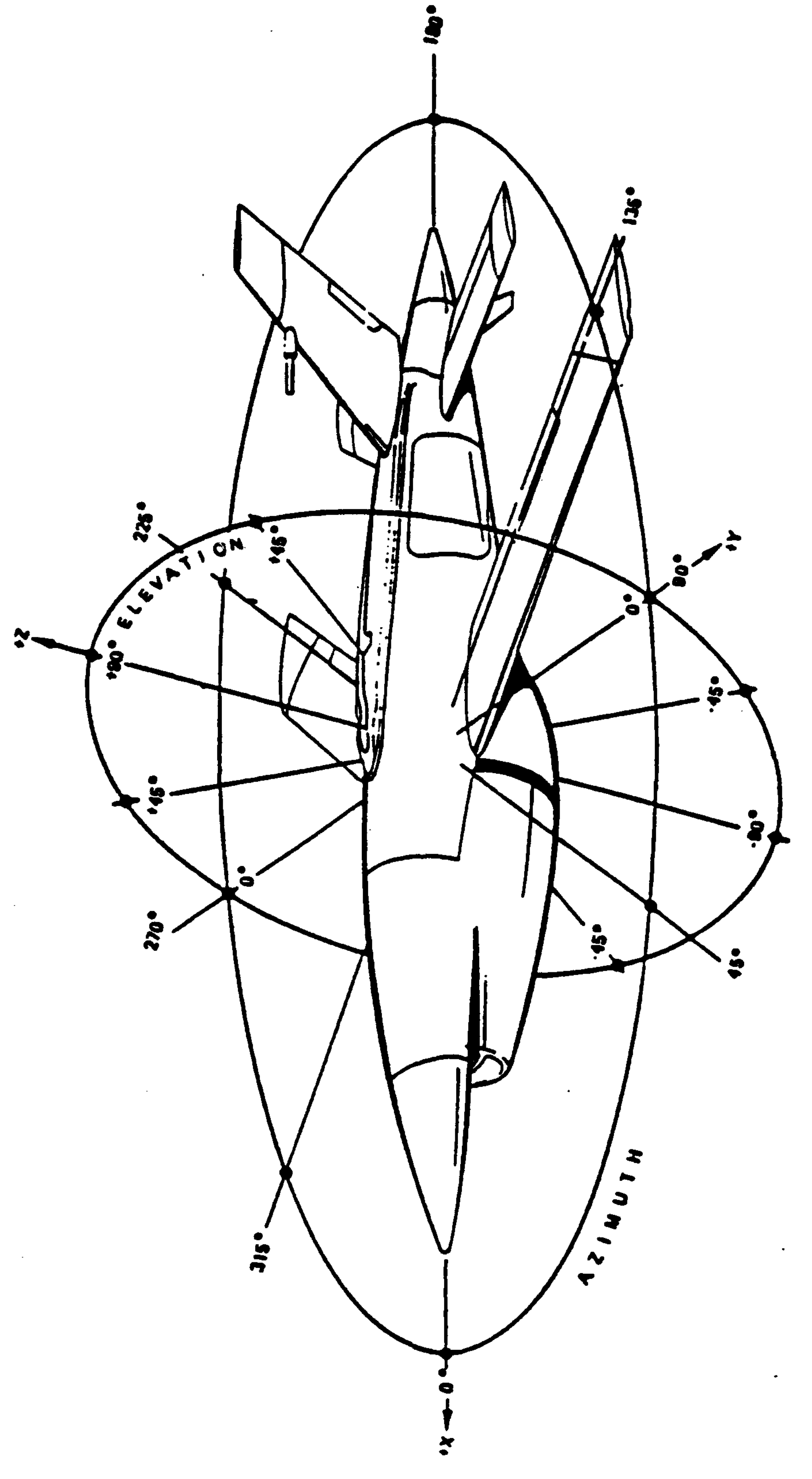


Figure 1. Illustration of Attack Aspect

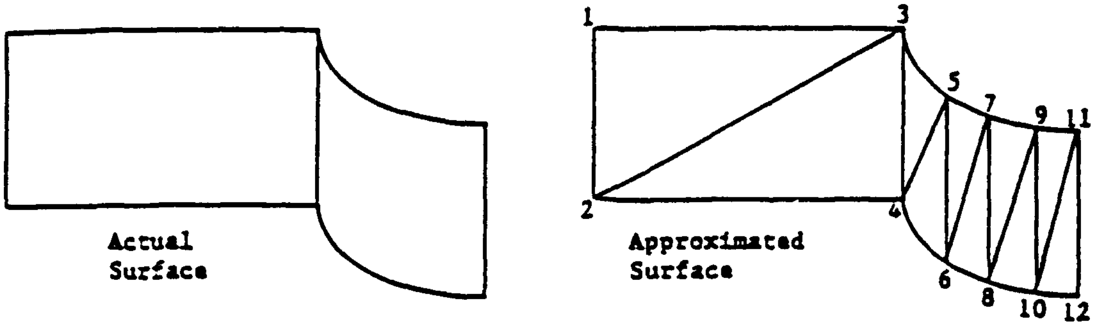


Figure 2. Approximations of Flat and Curved Surfaces Using Triangle Primitives

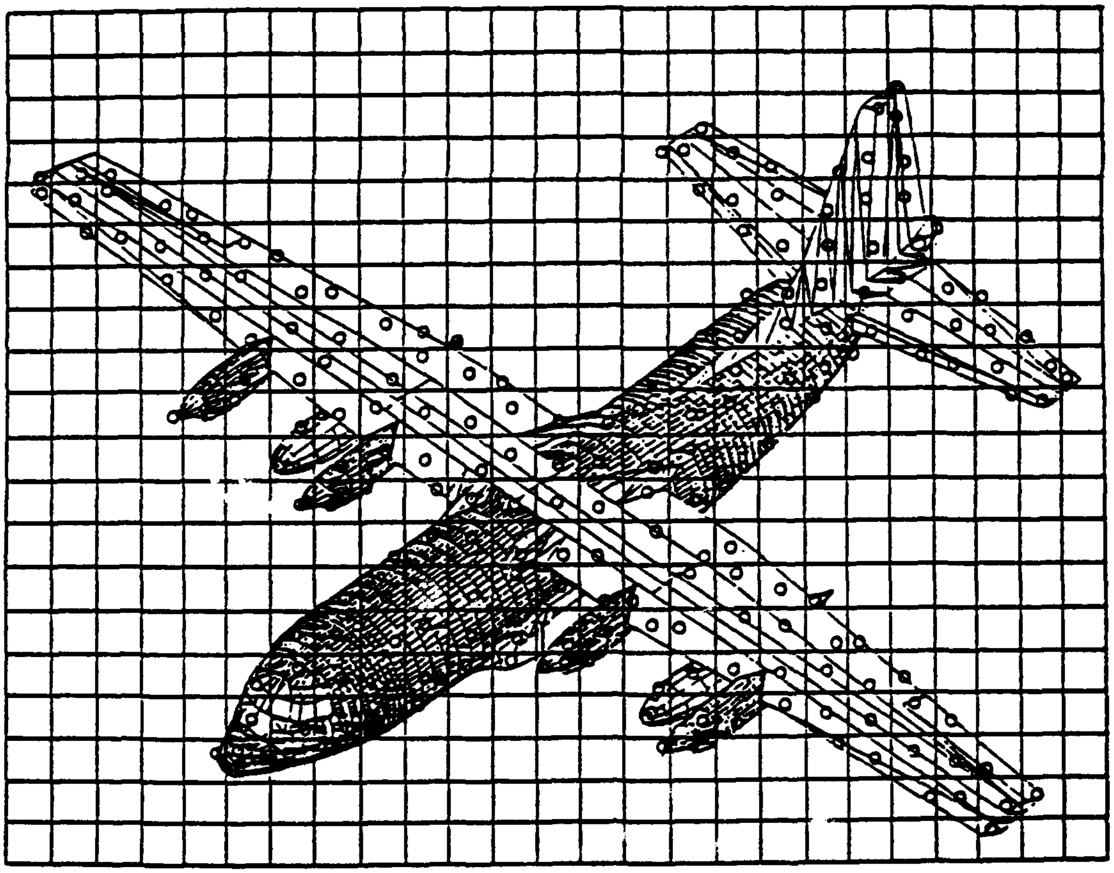


Figure 3. Grid Squares Normal to Attack Aspect

FASTGEN is a computer program which is a direct descendant of SHOTGEN and was also developed by Falcon Research. FASTGEN contains many modifications and enhancements to the basic methodology. The most important of these was the inclusion of certain geometric primitives having curved surfaces. For example, FASTGEN can compute ray intersections with spheres, and cylinders as well as with the triangles and influence mode (now called rod mode) components handled by SHOTGEN. Since 1976, FASTGEN has been upgraded with numerous improvements designed to enhance readability and maintainability, to facilitate program modifications, to extend program capacity, to reduce program generated errors, and to reduce computer resource requirements.

About the time FASTGEN was developed, Falcon Research developed a computer program called RAWGEN (Reference 5). Its purpose was to create a binary file from card image data. The resultant binary file was the basic target description data used as input to FASTGEN and some plot programs. Originally, RAWGEN was a very simple program consisting of a few lines of code to read the properly ordered data cards and create a binary file with 200 cards per record. Over the years, RAWGEN was expanded to accommodate new target description techniques and eventually became Program CONVERT.

2. PROGRAMS CURRENTLY USED IN VULNERABILITY ANALYSES

The CONVERT computer program is currently used to process target description data and to generate a binary file containing transformed target description data in the proper format for input to FASTGEN. Target geometric description data consists of three-dimensional coordinates which define components, component identification code names, space codes which identify areas or compartments where components reside, material thickness, and geometric codes which define the primitives (spheres, cylinders, donuts, triangles, boxes, wedges, and rods) used in modeling the surface of the target and target components. The FASTGEN formatted file is a binary blocked file with 200 records in each block. When this file is input to FASTGEN, the program produces line-of-sight (LOS) descriptions of the components and air spaces (or compartments) encountered by uniformly distributed parallel rays (shotlines) emanating from specified attack aspects. The file containing these descriptions is generally referred to as the LOS file. The target description data file produced by CONVERT and the LOS file generated by FASTGEN form the inputs to several vulnerability programs used to predict types and levels of damage to both ground and airborne targets. The relationship of CONVERT and FASTGEN to vulnerability and plot programs is illustrated in Figure 4. A brief description of each program is contained in the paragraphs that follow.

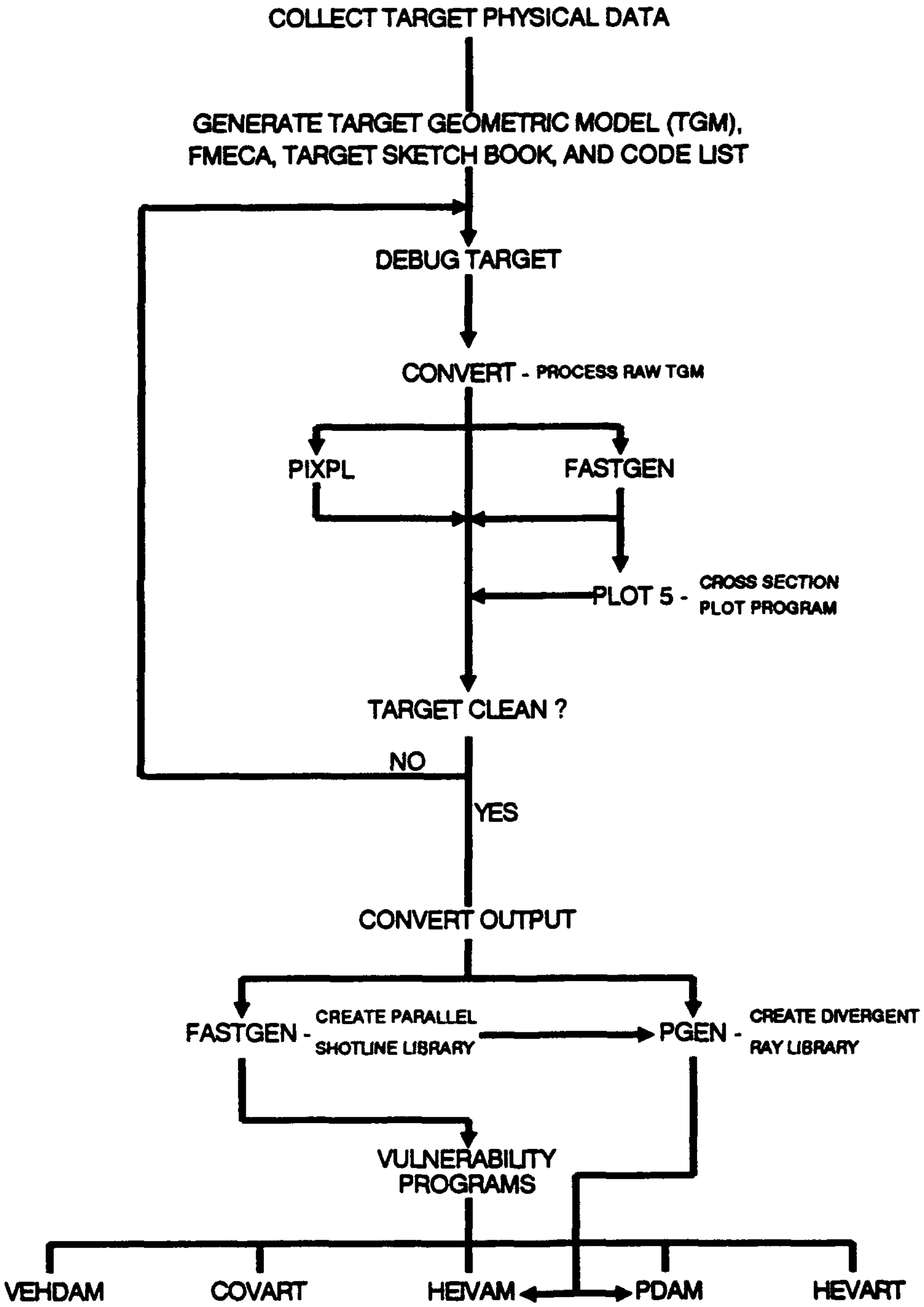


Figure 4. Vulnerability Program Family Relationships and Input Requirements

a. CONVERT

CONVERT is a versatile and flexible program which is used extensively in TGM development as well as to transform target geometric data. It allows, for example, shapes of primitives to be defined in simple terms. Spheres may be described by defining a center point, and radius. Similarly, cylinders (which includes truncated cones), donuts, and rods may be described by defining the coordinates of the end point centers and radii. Common shapes, such as boxes and wedges which otherwise would be described with triangles, may be described by defining the coordinates of four vertices. CONVERT transforms these shapes using triangle primitives with 16 sequenced points used to describe a box and 12 sequenced points to describe a wedge. While FASTGEN accepts component descriptions defined using triangle, sphere, cylinder, donut, and rod mode primitives, plot programs often plot only those components described using triangle and rod mode primitives. CONVERT provides an option to transform spheres, cylinders, and donuts to equivalent triangular approximations using triangle primitives as follows:

- Transform cylinder (including truncated cone) primitives and donut primitive using triangle primitives defined by 4, 6, 8, 12, 16, 24, or 32 points around each end.

- Transform sphere primitives using triangle primitives defined by 38 sequenced points.

CONVERT has several capabilities embedded in its logic to aid in describing components and to properly organize input data. It can, for example:

- Create mirror image component descriptions about the longitudinal axis of the target (In these cases, the description of one component is input and CONVERT reverses the Y coordinates of the original vertices to define the mirror image component).

- Create properly sequenced approximations of a component when input for the component contains multiple sequence numbers and a triangle vertex assigned on a single record.

- Sort records for components in their proper sequential order. In these cases, triangle vertex data can be recorded and records arranged in any order. CONVERT will then arrange the data for the triangle primitives in their proper sequence.

During execution, any one or all of the these options can be used. For example, if CONVERT is to be used to generate plots, all sphere, cylinder, and donut primitives must be converted to equivalent triangle primitives. If CONVERT is to be used to transform data for use in the FASTGEN program, and if mirror image components are to be created and/or if target description records contain more that one sequence number, CONVERT must be used to transform these data to a format acceptable as input to FASTGEN. FASTGEN will, however, accept triangle, sphere, cylinder, donut, and rod mode primitives and these need not be transformed. CONVERT always transforms boxes and wedges, described by four vertices, to triangles.

There are two input files used with CONVERT. One is the run specification file which specifies program options, and the other is the TGM deck which includes the target description data defining all of the surfaces of the masking and vulnerable components of the target. The primary output file from CONVERT is the binary blocked (200 records in each block) containing the transformed target description data. A printed output file is also generated with the amount of output depending upon the value of output options selected on the run specification file.

b. FASTGEN

The methodology used in FASTGEN is similar to that described earlier for SHOTGEN except that sphere, cylinder, and donut primitives, in addition to triangle and rod mode primitives, may be used to approximate component surfaces. As with CONVERT, FASTGEN requires two input files. One is the run specification file consisting of six records, three of which are mandatory. The second file contains the target description data formatted by CONVERT.

The primary output of the FASTGEN program is a binary blocked LOS file with eight words on each data record and 170 data records in each block. Two type of records are generated for the LOS file. The first record is written whenever a new attack aspect is processed and defines, for example, the azimuth and elevation angles of the attack aspect, the length and width of the grid cells through which shotlines are processed, and the maximum

and minimum values defining the limits used to enclose the target for the attack aspect being considered. The second record contains the descriptions of the components and air spaces encountered by each of the shotlines that pass through the target or, depending upon the option selected, by each of the shotlines which encounter critical components. A printed output file is also generated which echoes input data, provides definitions of the codes that are used to flag shotline errors, and displays error messages if applicable.

c. Point Burst Generator Program (PGEN)

PGEN (Reference 6) is a program designed to produce detailed descriptions of the components and air spaces encountered by rays emanating from burst points located inside or outside of a target. These detailed ray descriptions can be used as input to target vulnerability programs to determine the effectiveness of fragmenting and penetrating munitions used to attack the targets. PGEN can also be used to provide input to programs which predict the vulnerability of targets to behind-armor spall effects and to predict the effectiveness of small high-explosive (HE) munitions detonating externally or internally to aircraft structures.

A major option for selecting burst points is to use, as input, the shotline descriptions contained in the LOS output file generated by FASTGEN. Burst points are then selected along the shotline based upon warhead considerations such as the type of fuze for an HE warhead or an entry point into a critical region of the target for spall analyses. A required input to PGEN is the target description file generated by CONVERT. This may be the same file that was used as input to FASTGEN to generate the LOS file.

The principal output of PGEN is a binary file which contains such data as the burst point coordinates of the shotline, components and air spaces encountered by the shotline, and components and air spaces encountered by each ray.

d. Point Burst Damage Assessment Model (PDAM)

The PDAM (Reference 7) vulnerability program provides a methodology which models both the external and internal effects of anti armor munitions against tanks and fragmenting munitions against light armor vehicles and trucks. High-density kinetic energy (KE) penetrators, shaped-charge (SC), and HE warheads can be evaluated by PDAM. Multiple failure modes for each component can be evaluated using the most meaningful predictor of failure. For example, a meaningful predictor of damage to a suspension system component might be in terms of the total energy absorbed by the component. Fuel system

failure is best modeled on the basis of the size and location of holes resulting from debris. By using a functional failure mode fault tree to aggregate component functional failure, PDAM predicts the probability of achieving mobility, firepower, or catastrophic levels of damage for each weapon impact.

The type of analysis to be performed using PDAM is based upon any combination of three input files. One input file is the LOS output file generated by FASTGEN and the other two are output files generated by PGEN.

PDAM offers a selection of output formats which provide the analyst an understanding of the SC jet and fragment interactions with the critical components of the target and with data that may be used to verify the validity of the input data. The output selection is directed through four control flags. One pertains to printed output of input data and damage function tables, two pertain to printing of intermediate and summary results, and the last pertains to creation of plot output files used by separate plotting programs.

e. Computation of Vulnerable Area and Repair Time Program (COVART 3.0)

The COVART 3.0 (Reference 8) program is used to determine the vulnerable areas and expected repair times associated with specific levels of damage to targets caused by impacting single KE penetrators. Although emphasis is given to aerial targets, both fixed and rotary wing, vulnerable areas of ground targets can also be determined by COVART 3.0. Penetrators may be defined as fragments or projectiles which impact the target skin within a preselected weight and velocity matrix. Each penetrator is evaluated along each shotline and the contributions made along that trajectory to component and target vulnerable areas and to the repair effort are determined. The aircraft target velocity can be included when projectiles are to be evaluated. The weight and speed reduction for the penetrator is computed upon encounter with the surfaces of each target component along the shotline. Whenever a critical component is struck by the penetrator, the probability that the component is defeated is computed using input conditional probability of kill data. These data express the component kill probabilities as functions of threat impact (weight and speed). The component defeat probabilities are then combined, according to the various target damage definitions, in order to produce the target defeat probabilities for the given threat. When repair time data are to be determined, repair time information for damaged components, and combinations of components, are calculated (subject to the condition that the target survives to be repaired).

A required input to COVART 3.0 is an LOS file such as that generated by FASTGEN. Also required is an association table which describes the physical and functional relationship of and between components.

f. High-Explosive Incendiary Vulnerability Assessment Model (HEIVAM)

HEIVAM (Reference 9) is used to predict aerial target damage, in the form of vulnerable areas, resulting from an attack by small, high-explosive incendiary (HEI) projectiles employed under a wide variety of attack conditions. Weapon characteristics such as fusing type (contact or delayed), striking velocity, diameter, and weight may be varied in HEIVAM to cover a comprehensive set of inventory and developmental munitions. HEIVAM predicts whether a projectile will function normally, function partially, or ricochet. Fragmentation data (i.e., the number, location, and velocity of fragments produced when the projectile detonates) for a projectile are required inputs to HEIVAM. As fragmentation data may have been collected under static (zero projectile velocity) or dynamic (service projectile velocity or typical non zero striking velocity) test conditions, HEIVAM permits either type of data to be input. If static data are input, HEIVAM will calculate dynamic fragmentation data. If dynamic fragmentation data is input, no dynamic shift calculations are made.

Before executing HEIVAM, several other programs must be executed. These include CONVERT to process the BCD target description file in FASTGEN format, FASTGEN to generate the LOS file containing shotline descriptions, and PGEN to generate the point burst library. PGEN uses the output file of CONVERT and FASTGEN and user-supplied data to locate and correctly position the burst points along the shotline. and to create the burst point library. The burst point library, a file containing run specifications and weapon fragmentation characteristics, and a file containing the target functional and physical description data are used as inputs to HEIVAM.

g. PIXPL

PIXPL (Reference 10) is a program which provides a method of plotting projections of any TGM, or components of a TGM, which are described in a FASTGEN formatted input file. A condition exists, however, that cylinder, sphere, and donut primitives must be transformed by CONVERT as PIXPL will only accept components described using triangle and rod mode primitives. PIXPL is capable of producing both orthographic and perspective plots of the target for specified viewing angles around the target. In orthographic projections, the viewing point (observer's eye) is considered to be at infinity, and the visual

rays are parallel to each other and perpendicular to the projection plane. In perspective, the observer is considered to be at an finite distance from the object and rays extend from the observer's eye to all points of the target.

PIXPL has a number of options which allow considerable flexibility in the selection of TGM components that are to be plotted. Components can be plotted individually or by groups (i.e., all the fuel system components) or by groups with specific components within the group being excluded. Also, components that are not in a group of components selected to be plotted may be added individually. Another option identifies components that are to be defined as transparent components such as an aircraft canopy or a vehicle window. This option allows the user to plot the transparent component and those components that otherwise would be hidden by the transparent component. Finally, there are options in PIXPL which allow clipping of the target and target components. Clipping occurs by defining one or two planes through the target which, depending upon the options selected, removes from plotting all components or parts of components physically located either in front of the front clipping plane, or behind the back clipping plane.

Plotting devices used with PIXPL include the Calcomp 1039 with 921 Controller or Calcomp 1055 with 909 Controller, the Stromberg Carlson SC-4020, the Tektronix 4014-1/4054 using a vendor supplied preview package, and the Postscript graphics option.

h. PLOT5

PLOT5 (Reference 11) is a computer program used primarily as a target debugging aid. The program was developed to generate plots of the shotline data produced by a parallel ray tracing program, such as FASTGEN, that produces LOS data in a PLOT5 compatible format. Essentially, the program draws the side view of a ray. This is done by drawing a line to a prescribed scale and related to the plotter coordinate system to represent that portion of the ray that intersects a target component. The pen is lifted and moved the appropriate scaled distance over air spaces. A series of such line segments, for successive rays in the same plane, constitutes a cross sectional representation of the target.

An existing LOS file from a production run may be used as input to PLOT5, or a new LOS file can be made with rays restricted to the desired cross sections, by the use of the FASTGEN envelope option, and with a small grid cell size to provide enhanced detail. The envelope option is desirable if a very small grid cell size is used, otherwise a very large LOS file can result with a large target.

i. High-Explosive Vulnerability Area and Repair Time (HEVART)

HEVART (Reference 12) is used to predict aerial target damage, in the form of vulnerable areas and repair times, resulting from an attack by small HE/HEI projectiles. Weapon characteristics such as fuze type, striking velocity, projectile weight/diameter/length and fragment characteristics may be varied to cover a comprehensive set of munitions. HEVART uses static fragmentation data and projectile velocity to vector the warhead down the shotline until it detonates, computing the residual velocity after it penetrates each component. HEVART uses a preprocessor which sorts and arranges the shotline file generated by FASTGEN into a form which allows reconstruction of the target. This feature permits conical fragment zones to interact with components defined on the shotline file.

j. Vehicle Damage Computer Program (VEHDAM)

VEHDAM (Reference 13) is a computer program which computes vulnerable areas and kill probabilities of armored targets resulting from encounters with conical SC warheads. Program output is in terms of target kill probabilities and vulneraole areas for specified combinations of attack aspect and warhead charge diameter. The warhead kill probability, averaged over all azimuth angles, is also determined for each combination of kill category and elevation angle for each warhead processed.

3. TARGET MODEL DEFINITION AND DESCRIPTION

The initial requirement of the modeling preparation process is to define the purpose of the target model. An exact definition directly influences the number and type of components to be modeled and permits the establishment of realistic model preparation estimates. Target models are normally prepared to depict the vulnerability of components, which when damaged, cause some expected level of target incapacitation. This requires the description of both the vulnerable components and shielding components. Shielding components reduce the effectiveness of damage mechanisms by masking vulnerable components and absorbing warhead/fragment energy.

The first step in providing a description of a target involves the collection and close examination of all available information concerning the target. This, in itself, can be a difficult and arduous task. Engineering drawings coupled with the availability of the actual target are the most preferred sources of data. However, if a foreign target is to be modeled, neither engineering drawings nor the target item may be available. In such cases, the modeler will need to research intelligence data and rely on existing maintenance and operational manuals,

photographs, and other relevant publications. If a target is available for exploitation, resources can be conserved by using an exploitation team to examine, measure and photograph the physical item. To support such exploitation, Denver Research Institute (DRI) uses stereophotogrammetric techniques to characterize the components. Data reduction is accomplished in a semi-automated fashion using a three-dimensional plotter and microcomputer. Regardless of the means by which data are collected, the ultimate goal during this phase of the target description effort is to prepare an accurate set of drawings showing the interior and exterior configurations of the target as viewed from the front, rear, plan, and both sides. To prepare these drawings, data are needed on the exterior dimensions of the target, the location and configuration of all interior and exterior target components, the armor or shell thickness, and the number and location of the crew and other personnel.

4. MODEL LEVEL OF DETAIL

In the selection of components to be modeled, good judgment must be exercised based on an exact definition of the purpose of the target model. One could, by employing the techniques described in this document, develop a target description that is nearly an exact replica of the target surfaces. As vulnerability methodologies have become more sophisticated, the emphasis has been towards high fidelity target models that can be used with an entire suite of vulnerability programs. However, the additional time and cost involved in preparing a highly detailed geometrical model can be excessive and unnecessary. Therefore, a compromise should be established which is both practical and efficient, yet provides a computer model of sufficient detail for purposes of vulnerability assessment. Areas to consider when defining a target model include the type of kill to be analyzed, the type(s) of damage mechanism(s) expected to be evaluated, and the expected application of the target model in subsequent analyses.

5. TARGET TYPES

There are many classes of targets which are candidates for target descriptions in FASTGEN format. These include aircraft, ground mobile vehicles, water craft, communications centers, missile launch vehicles, radar tracking units, space vehicles, and fixed high value structures such as maintenance hangers, bunkers, and bridges. Some examples of targets modeled in FASTGEN format are illustrated in Figures 5 through 7.

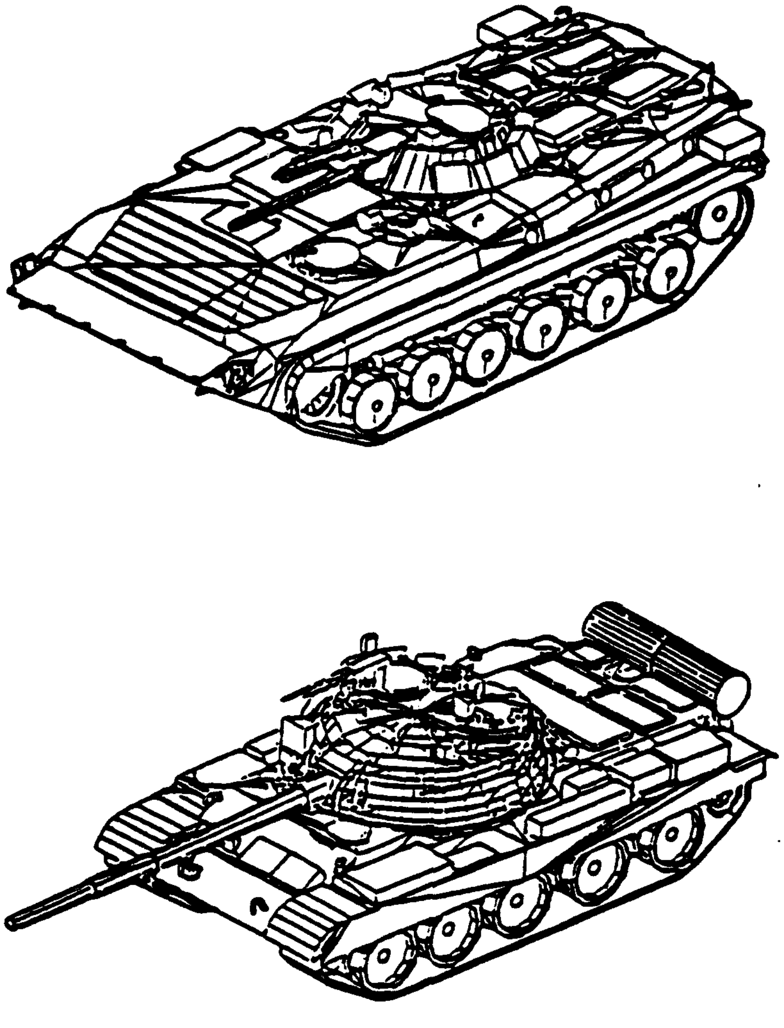


Figure 5. Ground Mobile Geometric Models

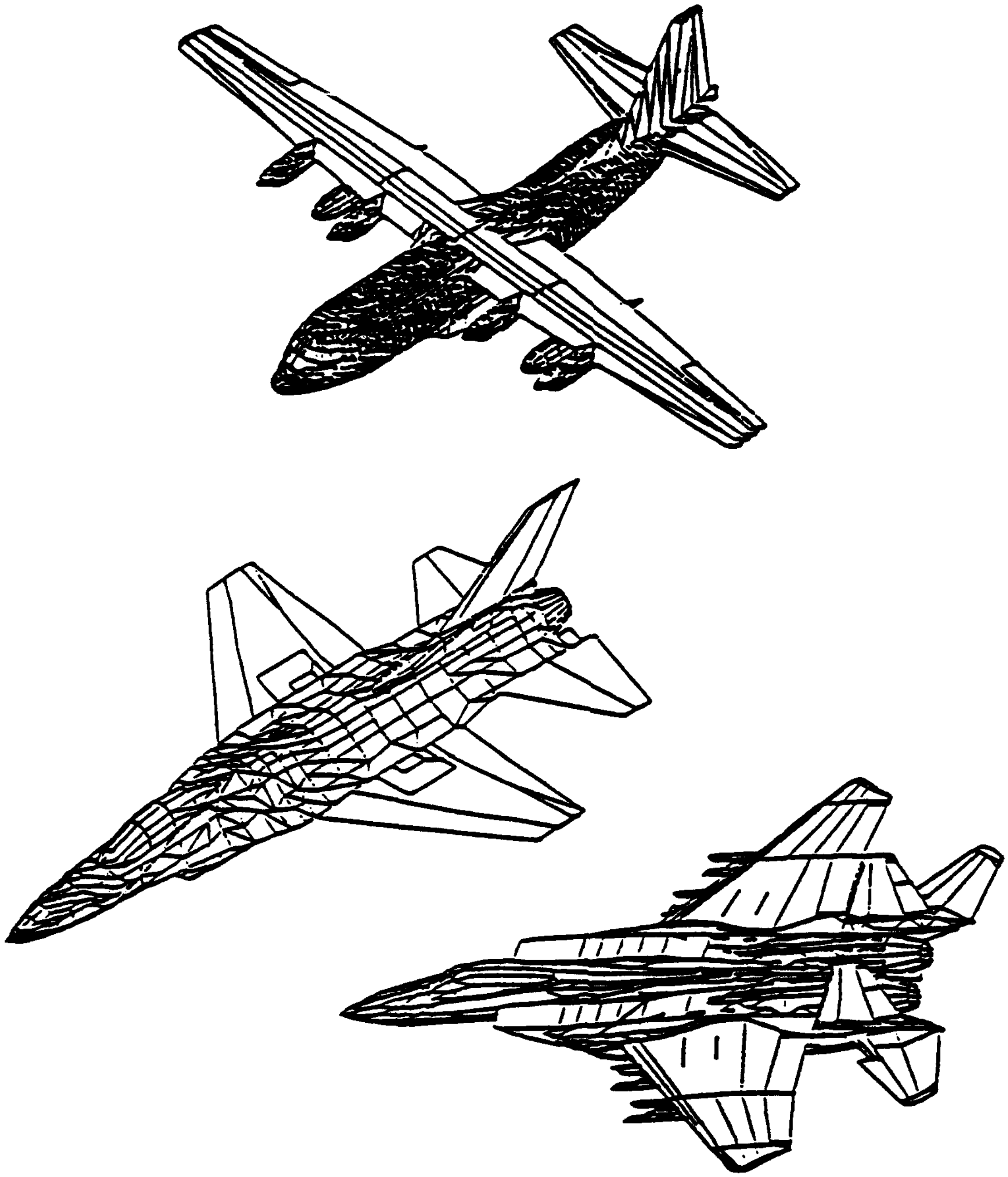


Figure 6. Aircraft Target Geometric Models

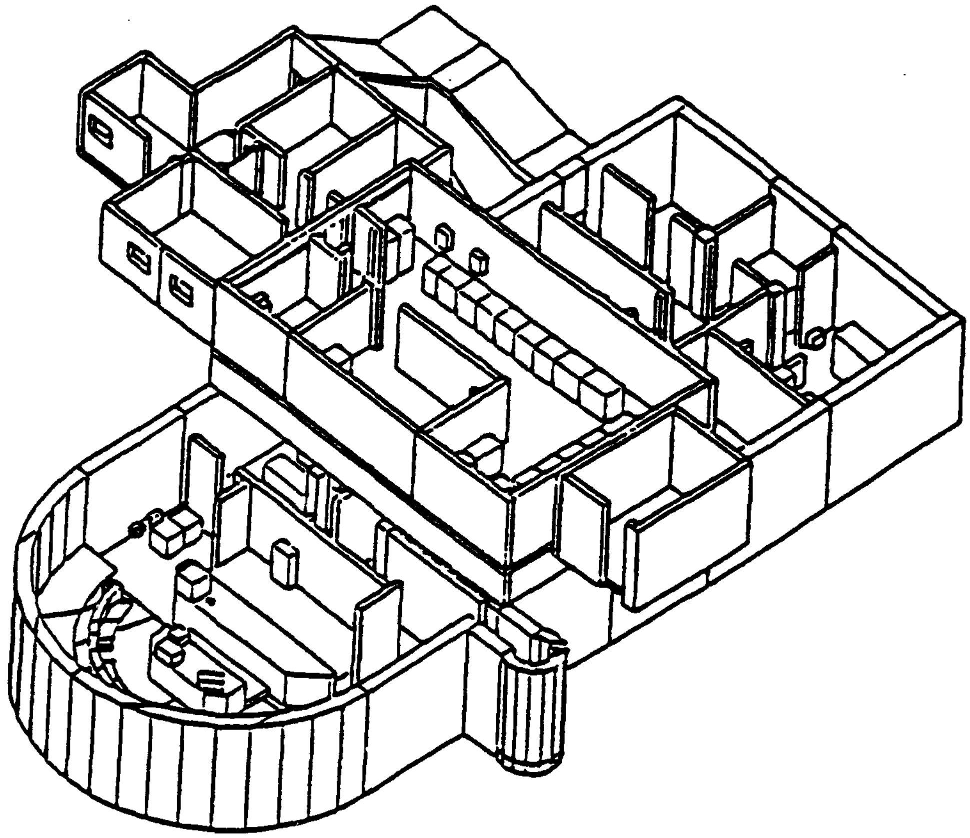


Figure 7. Communications Center Target Geometric Model

INTENTIONALLY LEFT BLANK.

SECTION II

FASTGEN MODELING CONVENTIONS

1. CONVERT CONSTRUCTS

The constructs used in developing a TGM consist of a selection of mathematical *approximations* to adequately describe the shape and surfaces of components which, taken collectively, define the target. Such approximations are in terms of seven primitives defined as triangle, cylinder (including truncated cone), donut, sphere, box, wedge, and rod mode primitives. With the exception of rod mode, CONVERT permits the description of each component using a combination of one or more, of the other six primitives. Hence, a single component may properly be described using one or more (or a combination of one or more) cylinders, spheres, boxes, and triangles. It is not necessary, however, that a component be modeled in a contiguous manner. That is, a single component can be modeled as separate parts, not physically connected to one another within the target model, but with the primitives of each part associated with a common component name. A more common practice however, is to break large complex components into sub-components to simplify the modeling process. For example, one would not normally model the turret of a tank as a single component. Rather it would be modeled in parts such as top, left side, right side, front, and back. Each part would be treated as a separate component and given component names such as Turret Top, Turret Left Side, Turret Right Side, Turret Front, and Turret Back. Similarly, one would not try to model a gun, an aircraft engine, or even a crew member as single component. The latter would be described with sub-component names such as Pilot Head, Pilot Left Arm, Pilot Right Leg, etc., with the head possibly being modeled using a sphere primitive, and arms and legs modeled with cylinder primitives.

Components described using rod mode primitives can not contain descriptions using any of the other primitives. Thus a component containing wires, lines, cables, control rods, and the like, must have these parts modeled separately and identified as individual components. Regardless of how a component is modeled, the entire component must be composed of only one material and density to allow proper component penetration treatment in subsequent vulnerability programs.

2. COMPONENT CODES

FASTGEN requires that each component (including sub-components) be identified by a unique integer not to exceed four digits; thereby distinguishing one component from another. These numbers are referred to as component code numbers. All primitives used to describe a given component must be assigned the component code number associated with that component. This convention allows, for example, shotlines generated by FASTGEN, which intersect the surfaces of primitives, to associate a component code number with the surface intersected. The assignment of component code numbers to components is not entirely arbitrary. The standard component code categories for ground vehicles and aircraft are listed in Tables 1 and 2 respectively, along with some examples of components which might be placed in each category.

TABLE 1. COMPONENT CODE CATEGORIES FOR GROUND VEHICLES

COMPONENT CODE NUMBER	MAJOR CATEGORY	EXAMPLES
0000 THROUGH 0999	ARMOR	Plates, hatches, turrets, engine cover, bulkheads
1000 THROUGH 1999	ENGINE AND ACCESSORIES	Cylinders, valves, crankcase, oil pump, oil lines, main bearings
2000 THROUGH 2999	CREW	Head, thorax, abdomen, arm, leg
3000 THROUGH 3999	PERSONNEL OR CARGO	Passengers, truck cargo
4000 THROUGH 4999	FUEL SYSTEM	Fuel pumps, tanks, lines, filters
5000 THROUGH 5999	AMMUNITION	Projectiles, propellant charges, ammunition boxes, grenades
6000 THROUGH 6999	ARMAMENT	Gun barrel, bore, breech block, recoil mechanism, trunnion
7000 THROUGH 7999	POWER TRAIN and SUSPENSION SYSTEM	Transmission, clutch, couplings, shift controls, parking brake torsion bar, tire, sprocket, tracks, wheels, housings
8000 THROUGH 8999	ELECTRICAL	Voltage regulator, generator, wires batteries, fuse box
9000 THROUGH 9999	MISCELLANEOUS	Air compressor, preheater, spare parts

TABLE 2. COMPONENT CODE CATEGORIES FOR AIRCRAFT

COMPONENT CODE NUMBER	MAJOR CATEGORY	EXAMPLES
0000 THROUGH 0999	SKIN AND OTHER EXTERNAL COVERS	Nose cone, wing, rudder, flaps, canopy, fuselage, gear doors
1000 THROUGH 1999	POWER PLANT AND ACCESSORIES	Inlet guide vanes, rotor blades , stator, gear box, oil pump and tank, fuel pump, oil cooler, generator
2000 THROUGH 2999	CREW	Head, thorax, abdomen, arm, leg
3000 THROUGH 3999	FLIGHT CONTROL SYSTEM & HYDRAULICS	Control stick, trim, flight control actuators, rudder pedal, linkage and rods, hydraulic reservoir and lines
4000 THROUGH 4999	FUEL SYSTEM	Pumps, lines, filters, tanks, fuel in tanks by levels i.e., 25%, 50%
5000 THROUGH 5999	AMMUNITION (INCLUDING BOMBS) & MISSILES	Missile warhead, body, fins, seeker, motor, pylon
6000 THROUGH 6999	ARMAMENT	Ammo drum, gun barrel, feed chute, flares
7000 THROUGH 7999	STRINGERS, RIBS, AND STRUCTURAL MEMBERS AIRFRAME	Wing spars and ribs, frames, bulkheads, engine frame, support structures
8000 THROUGH 8999	FIRE DIRECTIONAL SYSTEM AND AVIONICS	Radar, flight instruments, ECM components, nav-aids, IFF, data link system, gun camera, intercom, electrical wiring
9000 THROUGH 9999	MISCELLANEOUS	landing gear, pilot's seat frame, seat, cushion, head rest and helmet

3. SPACE CODES

A requirement exists for each primitive of each component to be associated with a unique single digit integer (space code) identifying the compartment or region of the target adjacent to the surface of the primitive. Space codes are used to facilitate debugging a target model and to provide a means of altering vulnerability, depending upon the space surrounding the component. They are also useful in plotting programs where the user has an option of selecting components to plot by compartments. All space surrounding a target is normally defined as exterior space (space code 5). All other space is interior to the target and must be defined by the modeler. Typical space codes, used to define regions within vehicular and aircraft targets, are shown in Tables 3 and 4.

TABLE 3. TYPICAL SPACE CODES ASSIGNMENTS FOR VEHICLES

CODE NUMBER	TYPE OF SPACE
0	BULKHEAD (PLATE MODE ONLY)
1	ENGINE COMPARTMENT
2	CREW COMPARTMENT
3	CARGO COMPARTMENT
5	EXTERIOR

TABLE 4. TYPICAL SPACE CODES ASSIGNMENTS FOR AIRCRAFT TARGETS

CODE NUMBER	TYPE OF SPACE
0	BULKHEAD (PLATE MODE ONLY)
1	FUSELAGE AND ENGINE PODS
2	COCKPIT
3	INTERIOR OF WINGS, VERTICAL FIN, AND ELEVATORS
5	EXTERIOR

The primitives of components with component code numbers less than 1000 (armor, aircraft skin, radome, canopy, and other external coverings) are assigned space codes which define the space of a compartment or region within the target and adjacent to the surface of the primitives. Thus, if an aircraft canopy is being modeled and the modeler wished to define the region which the canopy is to enclose as the cockpit with a space code 2, then each primitive defining the interior surface of the canopy and all primitives defining components located within the cockpit must be assigned a space code 2. Similarly, if the primitives defining the interior surface of armor surrounding the engine compartment of a tank are coded with a space code of 1, the primitives of components within the engine compartment also must be assigned a space code 1.

It is not a requirement of either CONVERT or FASTGEN that the modeler adhere strictly to assigning space codes to compartments according to Tables 3 and 4. For example, there are several recently developed aircraft TGMs where the entire target, interior as well as exterior, was modeled using a space code of 5. The decision on whether or not individual compartments need to be identified by the assignment of space codes depends on the requirements of the vulnerability programs with which the TGM is intended to be used. For example, if HEIVAM is to be used to evaluate the performance of a contact or delay-fuzed HEI warhead against an aircraft target, the identification of compartments is not normally required. Conversely, if a KE warhead is to be evaluated by PDAM against an armored vehicle, PGEN needs to establish at what point, along the shotline, perforation of armor into a critical region occurs in order to establish a burst point for spalling purposes. This is accomplished by reference to a space code defining entry into a critical region.

4. DESCRIPTION MODES

The triangle, sphere, cylinder (including truncated cone), box, or wedge primitives used to define a component must be described in one of two modes: plate or volume. Plate mode is used to describe a surface and a normal thickness. For example, the skin covering the surface of an aircraft wing may be described using triangle primitives to define the outer surface and a material thickness to define the normal thickness of the skin. Another example of modeling in plate mode is to consider a simple fuel tank that may be modeled by defining it as a box (using four vertices) and defining the thickness of the box walls. The advantage of describing these two components in plate mode is that neither the inner surface of the aircraft wing nor the inner surface of the fuel tank need be described. A limitation is that plate mode

may only be used for components or component parts which range in thickness from 0.01-inch up to and including 2.99-inches and is generally recommended for components having a normal thickness of 0.50-inch or less.

The volume mode is specified when all surfaces of a component or parts of a component are described. An example of a component modeled in volume mode is the turret of a tank where the thickness of the armor may vary from a few inches to more than 10 inches. In this case, both the outer surface and the inner surface of the armor would be modeled. The armor thickness would not have to be specified.

In general, all components may be modeled as a series of plates or a series of volumes, however caution should be exercised to ensure the correctness of the computer model for the intended application. The mode selected should be determined by the type of assessment to be conducted. The easiest method of modeling a component may not be the most appropriate. Additionally, the incorportion of predefined shapes or primitives should be used to simplify the modeling process. In some instances, the use of predefined shapes will not "look" correct when graphically displayed, although for the purposes of the vulnerability assessment, the shapes are acceptable. The modeler should be aware of some of the appropriate graphics routines and create the computer model accordingly. For example, an aircraft intake modeled as a cylinder will be drawn with the ends closed which can be handled correctly with the vulnerability codes but will not look correct when drawn. A solution to this problem would be to define the intake as a donut or as an irregular shaped object with each triangle defined by the modeler.

Although mixing plate and volume modes within the same component is allowed, this practice should be avoided whenever possible. Future geometric model translations may have difficulty dealing with such complex components.

5. DESCRIPTION CODES

The description code is a packed integer which specifies whether the primitive used in defining a component is described in the volume, plate, or rod mode, the type of geometric primitive used to approximate the surface, the normal thickness (if the description is in the plate mode), the radius of the component (if it is described in the rod mode), and the space code for the air space or compartment immediately adjacent to the surface being defined.

When a component or component part is described in either volume mode (+) or plate mode (-), the form of the description code is + or -GNNS, where G is the code number which

defines the primitive type, NN is the normal thickness (in inches x 100) of the component, and S is the space code. Table 5 shows the code numbers defining primitives. If a component or component part is to be modeled in volume mode, the plus sign (which is optional and usually omitted) applies and the value entered for NN is usually 00 as FASTGEN calculates the LOS thickness of a volume mode component based upon the location and orientation of the entrance and exit surfaces of the component. The exception to this is that a modeler will often enter the thickness of a volume mode component when modeling volume armor (component number less than 1000) in order to aid in the debugging process. Primitives modeled in volume mode will appear to FASTGEN and vulnerability programs as solids.

For components or component parts modeled in plate mode, the minus sign (which must be entered) applies and a value entered for NN defining the normal thickness in inches times 100. Cylinders, spheres, boxes, and wedges will appear hollow with wall thicknesses defined by the value placed in NN. For example, if a hollow cylinder having a 0.25-inch wall thickness is being modeled, a value of 25 (0.25 times 100) would be entered for NN.

If a component or component part is being modeled using triangle primitives, the G of GNNS will be treated as an N yielding a description code which one may interpret as NNNS. Since the maximum value of G for triangle primitives is 2 (as shown in Table 5), then the maximum thickness allowed by the NNN is 299 (2.99 inches). CONVERT however, allows a maximum of 32 components, defined with triangle primitives, to be described with a normal thickness greater than 2.99 inches. This is accomplished by inserting the thickness (in inches x 100) in the GNN portion of the description code, and then defining the component code number in the TGM deck. For example, if a component described in volume mode has a thickness of 5.4-inches and the modeler wished to record this thickness in the description code, 540 would be entered the GNN portion of the code and the component code number defined in the TGM deck so that the 5 would not be interpreted as a G code defining a wedge. There is nothing in either CONVERT or FASTGEN which prevents one from similarly modeling, and defining in the TGM deck, a thick plate mode component modeled with triangles (-GNNS). However, the modeler is advised that modeling thick plate mode components can lead to serious modeling errors (described in the next section) and the practice should be avoided. Modeling in plate mode should be limited to those components having a wall thickness of not more than 0.50 inch (preferably less than 0.25 inch).

It is a requirement that adjacent interior regions of a target be separated by a component having a component code number less than 1000. The component may be real or imaginary.

TABLE 5. PRIMITIVE CODE NUMBERS

CODE NUMBER (G)	PRIMITIVE
0, 1, 2	Triangle
3	Donut
4	Cylinder (24 point)
5	Wedge
6	Sphere
7	Box
8	Cylinder (12 point)
9	Rod Mode

If the component is real and irregular in shape, it is usually modeled using triangle primitives modeled in volume mode. In this case, the space codes associated with the primitives forming the surfaces of the component would be those corresponding to the space code of the adjacent regions. If a real component is relatively thin and can be modeled in a plane; then plate mode may be used with NN of the description code equal to the normal thickness (in inches x 100) and S equal to 0. If no component separates the two regions, then an imaginary bulkhead should be modeled. This bulkhead should be modeled in plate mode with NN of the description code equal to 01 and S equal to 0.

When components are described in rod mode, the form of the description code is -9NNS, where -9 specifies that the component is described in the rod mode, NN is the radius of the component (in inches x 100), and S is the space code. The description code for rod mode components is always preceded by a minus sign.

Components with component code numbers less than 1000 are those that completely shield the interior volume of the target. For armored vehicles, these components may consist of heavy plates, sheet metal, grilles and louvers. For aircraft, similarly coded components include the skin covering wings, fuselage, and horizontal and vertical stabilizers, the radome, canopy, and gear doors. Since interior regions or compartments must be totally enclosed, openings such as vision ports on armored vehicles or tailpipes and air intake ducts on aircraft must be covered to separate an interior region from the exterior region. This is accomplished by creating phantom components. Phantom components are described in plate mode using triangle primitives where the GNN of -GNNS is defined as 000. and the S is the space code of the interior region.

Figure 8 shows the format and defines the parameters used in target geometric modeling. Figure 9 illustrates a form that may be adopted on which parameter values may be entered.

PARAMETER	UNITS	FORMAT	COLUMNS	DESCRIPTION
X(N1) Y(N1) Z(N1)	Inches Inches Inches	F8.2 F8.2 F9.2	1-8 9-16 17-25	Entries vary depending upon the geometric shape used to approximate the component surface. These entries can represent the X, Y, and Z coordinates of triangle vertices, the ends of truncated cones or cylinders, or the sphere center. Z(N1) can also specify whether one or both ends of truncated cone or cylinder are open or closed. Detailed instructions for entering these inputs are given in the text of the next section
IT(N1)	None	I6	26-31	A code number in the form of plus (+)GNNS or minus (-)GNNS, where: the plus sign (usually omitted) specifies that the component or component part is approximated using volume mode. A minus sign must be entered if the component or component part is described in plate mode. These rules apply to all primitives with the exception of rod mode primitives which are always accompanied by a minus sign. G specifies the code number of the geometric primitive used to approximate the component or component part. For triangles, the G represents the whole number part of inches. The G code numbers are defined as follows: 0, 1, or 2 = approximated with triangles. 3 = approximated by a donut. 4 = approximated by a 24-point cylinder. 5 = approximated by wedge 6 = approximated by sphere. 7 = approximated by box. 8 = approximated by 12-point cylinder. 9 = approximated in rod mode. NN specifies the normal thickness of the component (in inches x 100) when plate mode is specified or the radius of the component (in inches x 100) when rod mode is specified. When volume mode is specified, enter zero. S specifies the space code for the air space immediately adjacent to the component..
ICO (N1)	None	I4	32-35	The four-digit component code identification number.
ISQ (1) ISQ (2) ISQ (i) ISQ (8)	None None None None	I11 I4 5I4 I4	36-46 47-50 51-70 71-74	The sequence numbers that are assigned to the triangle vertex (maximum of eight). (i= 3 , 7)
MIRROR	None	I3	75-77	The value to be added to the component code identification number to create a mirror image component about the longitudinal axis of the target or, -1 to keep same component number.
IV	None	I3	78-80	The target identification number.

Figure 8. Definitions and Format for Target Description Data

X COORD OR RADIUS	Y COORD OR RADIUS	Z COORD	±GNNS	COMP CODE	SEQUENCE NUMBER	ADDITIONAL SEQUENCE NUMBERS							MIRROR	VC
00000000 12345678	01111111 90123456	111222222 789012345	222233 678901	3333 2345	33334444444 67890123456	4445 7890	5555 1234	5555 5678	5666 9012	6666 3456	6667 7890	7777 1234	777 567	778 890

Figure 9. Illustration of Component Coding Form

SECTION III

CREATING THE TARGET MODEL

1. MODEL PREPARATION

The first step in the preparation of a TGM is the selection of a coordinate system. The next two steps require the creation of a component code list and component sketches. While the first step must be completed at the outset of model development, the second two steps may be completed prior to, or concurrent with, the actual model development. Each step is described in detail in the following paragraphs.

a. Target Coordinate System

If a target is placed in a three-dimensional coordinate system, it is possible to define any point on the target in terms of that coordinate system. The system used in developing target models in FASTGEN is a right-hand Cartesian coordinate system. The location of the origin of the target coordinate system may be defined anywhere within or near the target. For example, the origin may be located at the approximate geometric center of the target coincident with some predominate and easily recognized feature of the target. For aircraft, the origin is normally located at the nose along the fuselage longitudinal centerline to take advantage of structural and system symmetry as illustrated in Figure 10. For ground vehicles, the origin is normally selected to coincide with the target centroid projected vertically on to the ground plane as in Figure 11.

b. Component Code List

The level of detail in the target description, needed to satisfy the exact purpose of the model, will govern the number of target components incorporated into the model. The target component code list reflects this level of detail and serves to organize component descriptions numerically into their respective subsystems.

Component code list entries can be developed effectively from target lethal criteria analyses. These analyses, referred to as FMECA (failure modes effects and criticality analyses) are developed from the physical and functional characteristics of both critical and non-critical components of the target. The FMECA will provide clues as to how detailed a target should be modeled depending upon kill level.

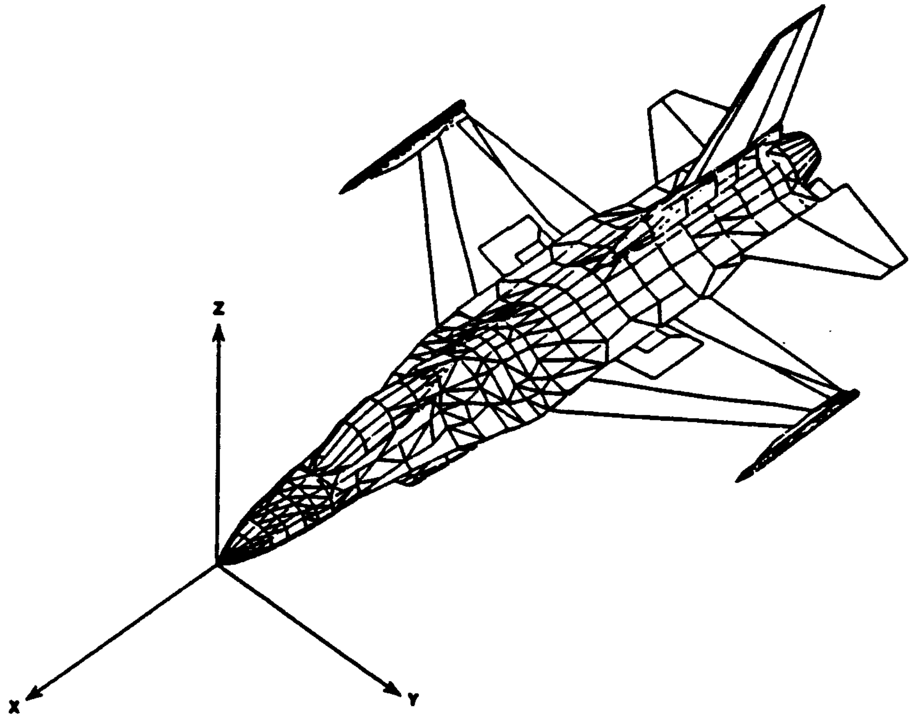


Figure 10. Coordinate System for Aircraft

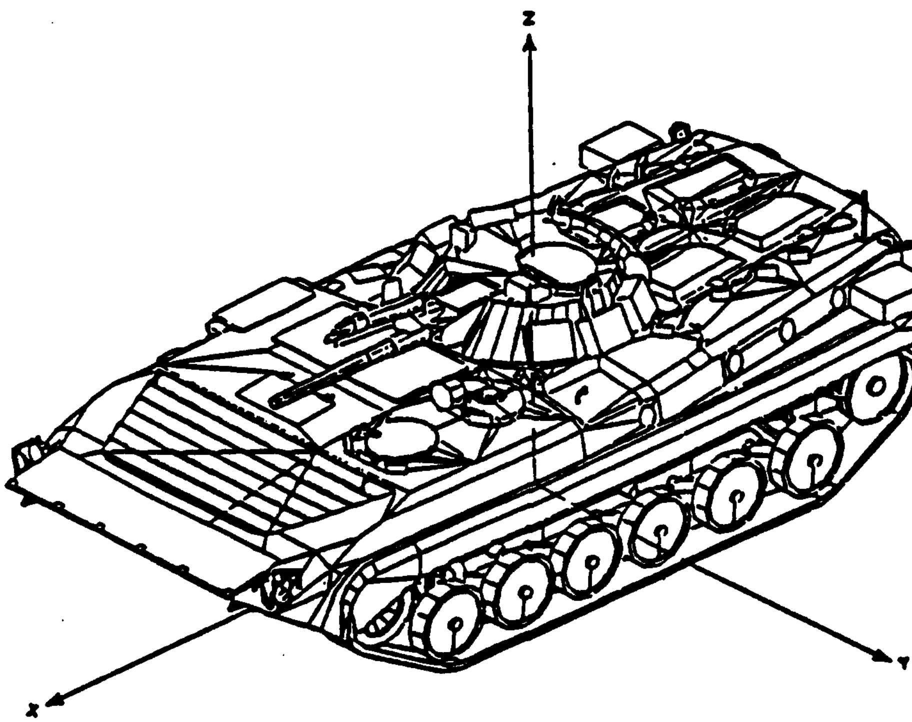


Figure 11. Coordinate System for Armored Vehicle

The standard code categories, shown in Tables 1 and 2 of the previous section, have been established for previous assessments of ground vehicles and aircraft. While each component must be placed within one of the major categories, a particular component may be specified in any one of the several code categories. For example, an exhaust manifold of a naturally aspirated internal combustion engine may be identified by a code number in the 9XXX series instead of the 1XXX series.

A convention found convenient by many modelers is that of creating sub-groups and sub-sub-groups within the major categories. For example, assume three crew members of an armored vehicle are to be modeled. Assume also that each crew member is wearing a protective helmet. The modeler might assign code identifiers such as 2110, 2111, 2112, 2113, 2114, 2115, 2116, 2117, and 2118 to the head, thorax, chest, abdomen, pelvis, right arm, left arm, right leg and left leg, respectively of crewmember number 1. Similarly, the modeler might assign 2210 through 2218 and 2310 through 2318 to the same components of crew member 2 and 3, respectively. Since the helmets are associated with the crew but are non-critical, the modeler could place them in the miscellaneous category (92XX) and assign code identifiers such as 9210, 9220, and 9230 relating the second digit (2) of the helmet code identifiers to the 2XXX component functional category and the third digit (1, 2, and 3) to the crew member.

Although there is no established format for a component code list, as a minimum a code list should contain for each component, the component code number, the component name, the code number from COVART 3.0 User Manual indicating the material of the component and the relative density of the material (Appendix A).

An example of the TGM deck including a sample component code list for aircraft is contained in Appendix B. Component number sequencing within major categories illustrates how component codes can be assigned in a logical and orderly manner. The final component code list is the result of an evolutionary process that transpires over the entire model-preparation process.

c. Component Sketches

The preparation of sketches of target components listed in the component code list is an integral step in target model preparation. All components should be consistently drawn as if the viewer were looking at the component from a point in space in front of and above its left front corner (45-degree azimuth, 30-degree elevation). Component circles,

which lie in planes that are perpendicular to the three coordinate axes, may be drawn with an isometric template. No attempt should be made to sketch the component to scale. Instead, the relative shape should be maintained, together with a minimum of interference between non-related component edge points. Clarity rather than precision should be the goal in component sketch preparation. Figure 12 illustrates a poor and a good sketch of the same component. The sketch labeled "b" provides more space for labeling primitives and vertices than does that labeled "a". The modeler will find it advantageous to adopt a standard worksheet format for sketch preparation similar to that used in Figure 12.

2. COMPONENT MODELING WITH PRIMITIVES

Execution of CONVERT requires two input files. One is the run specification file and the second is a file containing the target description data. The latter is a BCD file which defines all of the surfaces of masking and vulnerable components of the target. File records are grouped (dependent upon component type) and describe each individual component. A form, similar to that illustrated in Figure 9 of the previous section, should be used to record component data in the proper format. A detailed discussion of the rules for preparing the target description file, where components are described using primitives defined as triangles, boxes, wedges, cylinders (including truncated cones), donuts, spheres, and rods, is provided in the following paragraphs.

a. Triangles

Although many target components may be reasonably represented by regular geometric shapes such as cylinders, spheres, cones, etc., most targets contain components which are irregular in shape. FASTGEN permits those components which are regular in shape to be described exactly, while those components that have an irregular contour are described using triangle primitives.

A convenient mathematical form for describing a component surface is based on the fact that any surface, flat or curved, can be approximated by one or more flat triangular surfaces. Triangles are formed by connecting three non-collinear points, thus forming a plane. If more than one triangle is required to describe a surface, then the triangles must be sequenced such that any three successive points define a triangle (not already described) or a straight line on the surface of the component.

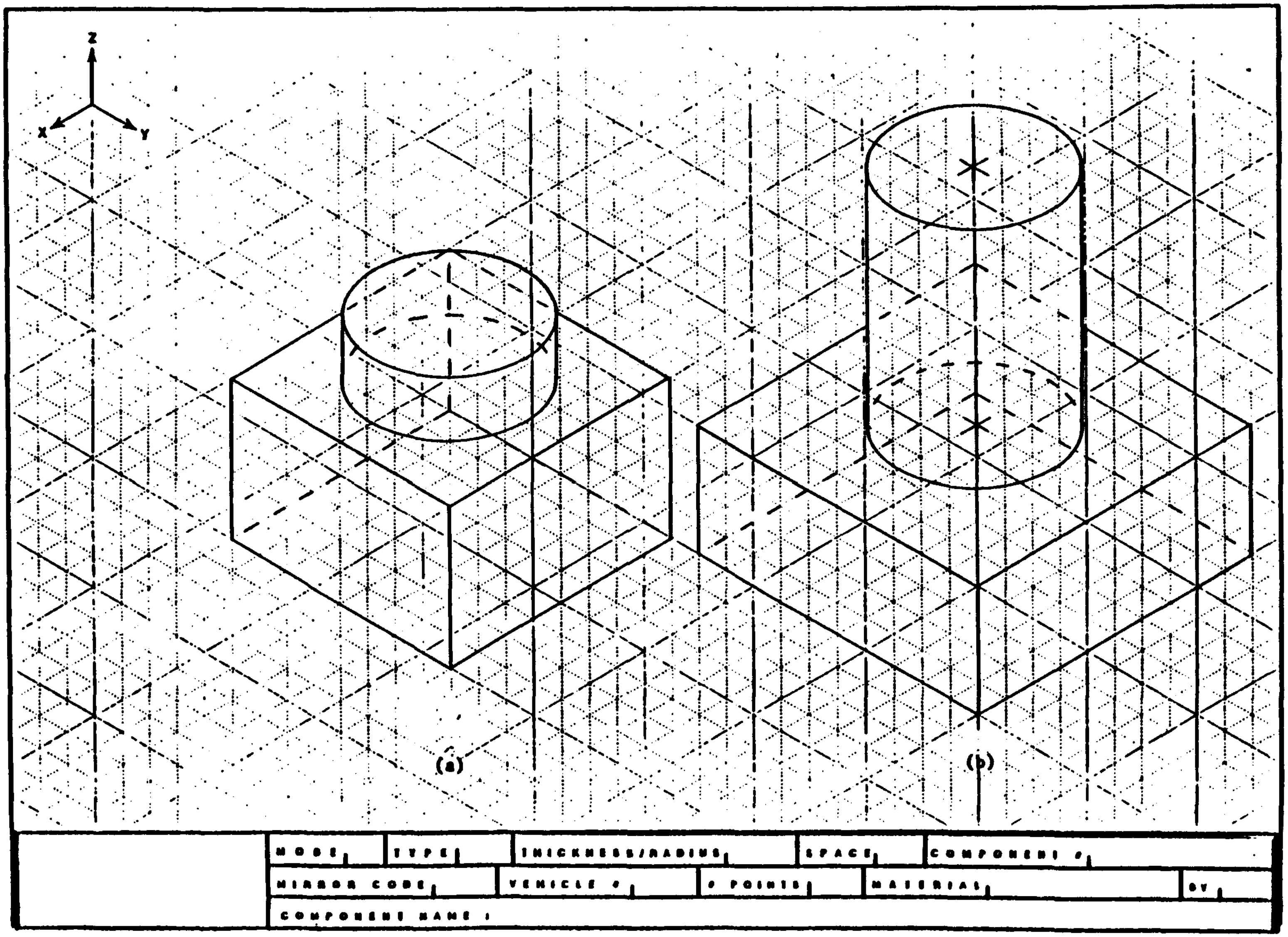


Figure 12. Comparison of: (a) Poor Sketch and (b) Good Sketch

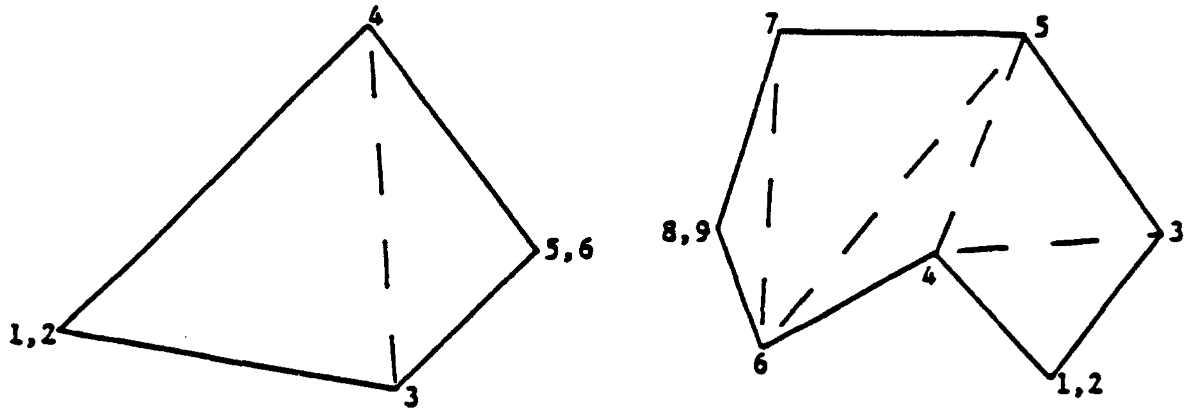
The proper selection of sequence points is critical. In Figure 13(a), points 2, 3, and 4 and points 3, 4, and 5 define two triangles which completely describe the surface of the quadrilateral. In Figure 13(b), points 2, 3, and 4, points 3, 4, and 5, points 4, 5, and 6, points 5, 6, and 7, and points 6, 7, and 8 define the five triangles which describe the entire surface of the polygon. If the sequence numbers in Figure 13 are not placed at their proper points, the entire surface of the geometric shape may not be defined and/or part of the surface may be defined twice. For example, Figure 14(a) illustrates an improper sequencing of Figure 13(a) since triangle A is described twice and triangle B is never described. In Figure 14(b), the sequence is improper because triangle A is described twice.

Note in Figures 13 and 14 that the first points of the first triangles and last points of the last triangles formed on the geometric figures are repeated points i.e., point 1 is repeated as point 2 and point 5 is repeated as point 6 on Figure 13(a), and point 1 is repeated as point 2 and point 8 is repeated as point 9 on Figure 13(b). Repeating the first and last points of components described using triangles is always required.

When describing two-dimensional surfaces that are not polygons (i.e., circles, ellipses, and other surfaces that contain curves), the surfaces may be approximated by considering them as polygons with sides which closely follow the outlines of the surfaces. Accuracy, of course, is dependent on the number of sides of the polygon used. Figure 15(a) shows an example used for approximating a circular surface, Figure 15(b) shows an example used for describing an irregular surface with both flat and curved sides, and Figure 15(c) shows an example used for approximating a curved surface.

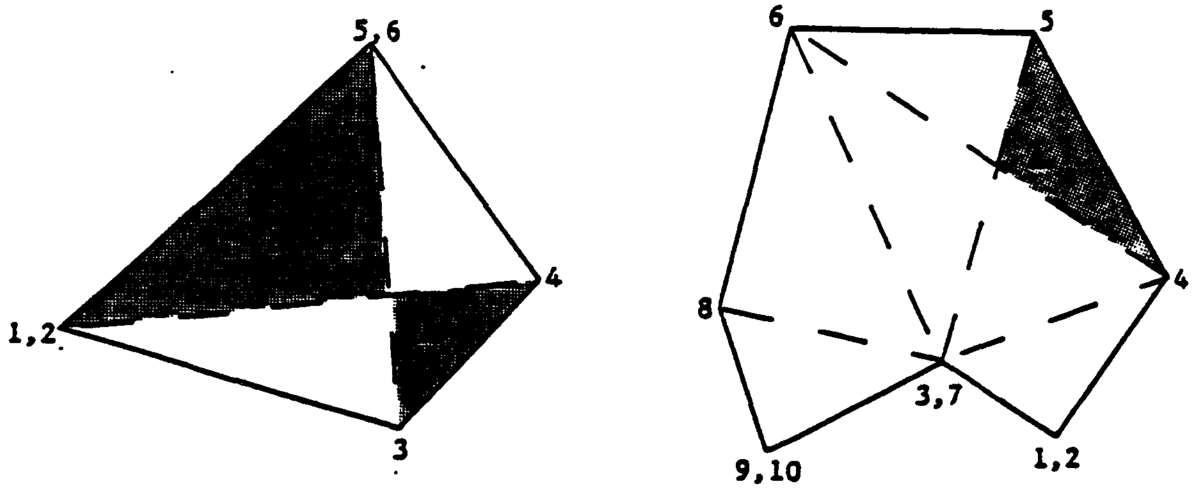
Surfaces of three-dimensional figures may also be described by a series of flat triangular surfaces. Flat-sided shapes will be exactly represented and curved objects will be approximated by a series of triangular surfaces. In describing three-dimensional figures by triangles, a repeat of coordinates will frequently occur. This duplication is necessary when repeated points are required to eliminate describing the same surface twice, or to properly begin describing another surface of a three-dimensional object.

The sequencing techniques that must be used in describing the surface of three-dimensional objects having flat-sides or curved surfaces are essentially the same as those used for describing plane surfaces. Figure 16 shows one acceptable method for sequencing these points for a three-dimensional flat-sided object. Note that the sequence of points in Figure 16 are chosen so that each surface of the object is described only once. Note also that a repeated



(a) (b)

Figure 13. Proper Sequence of Points for Flat-Sided Surfaces



(a) (b)

Figure 14. Improper Sequence of Points for Flat-Sided Surfaces

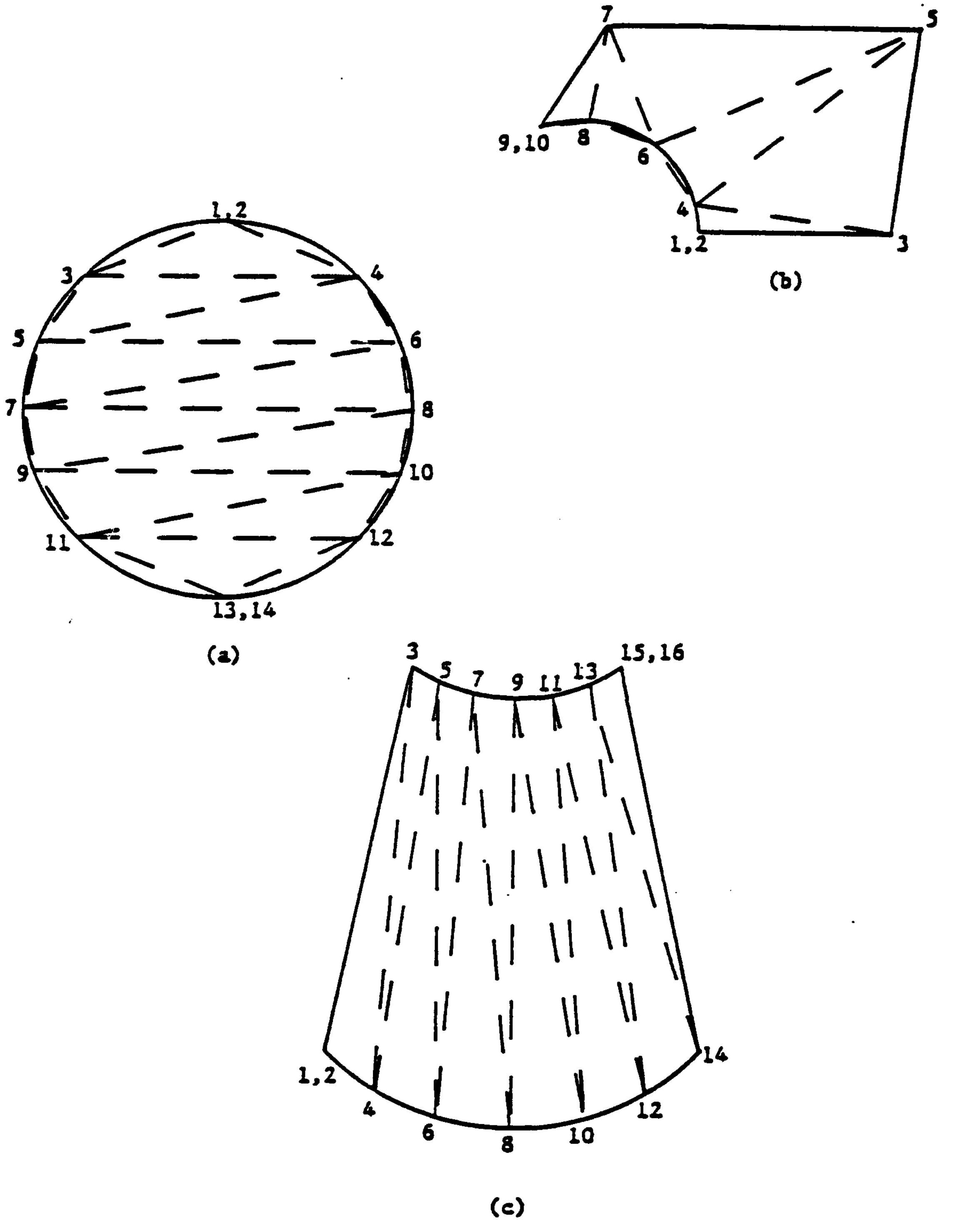


Figure 15. Approximation and Proper Sequencing of Surfaces Bounded by Curves and Curved Surfaces

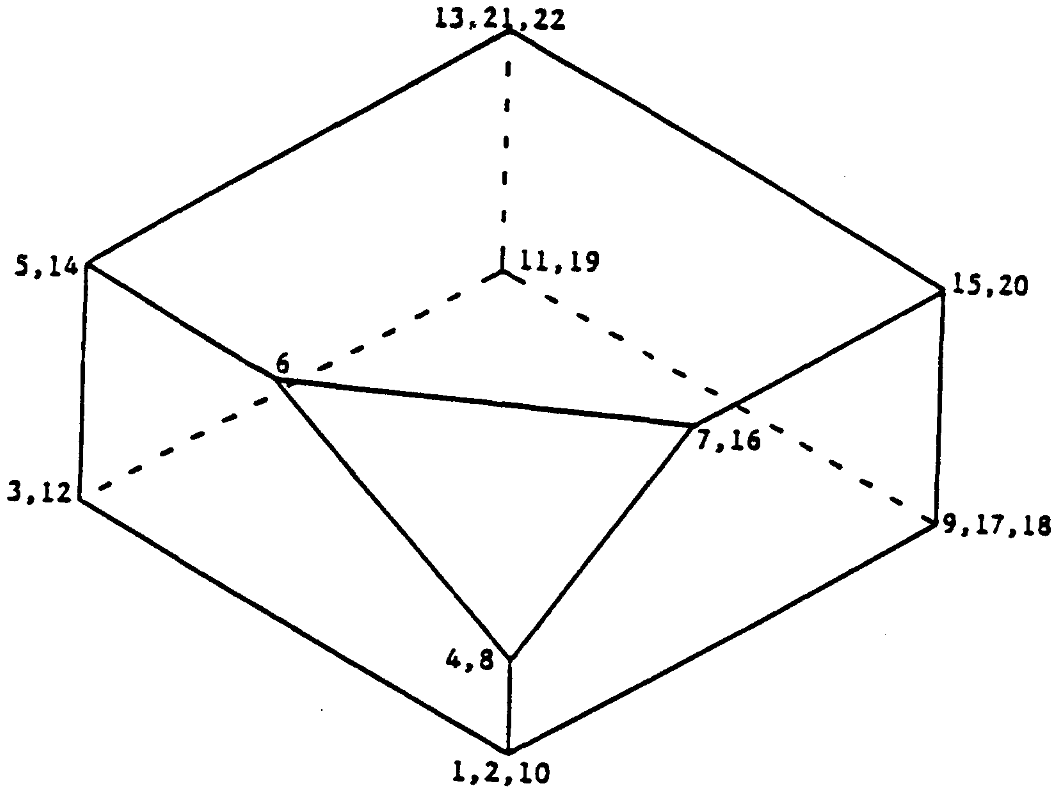


Figure 16. Point Sequencing of a Flat-Sided Object

point (defined by sequence numbers 18 and 19) is utilized so that the rear surface of the object can be described properly. If point 18 is not used as a repeated point but instead is located with point 11, then points 16, 17, and 18 will describe a triangle which does not lie on the surface. Point 19 is then placed so that points 17, 18, and 19 describe a straight line, which is the base of the next triangular surface to be described. Also, point 19 can be placed with point 15, and point 20 can be placed with point 11, and the rear surface of the object will be properly described. Figure 16 shows only one of many acceptable methods for sequencing the points of the object. Any method is acceptable so long as it describes the entire surface of the object only one time.

A sketch of a component defined using triangle primitives to describe its surface is illustrated in Figure 17. The component is modeled in plate mode and has a wall thickness of 0.12-inch. The sketch indicates the sequence points and below each set of points, the X, Y, Z coordinates of the point. The bottom of the sketch contains all the additional information needed for its description code (-GNNS). The minus sign (-) entered in the MODE block indicates plate mode, the 0 in the TYPE block indicates triangle primitives, the THICKNESS/RADIUS block contains the wall thickness (in inches x 100), and the SPACE block indicates the space code where the component resides. Figure 18 shows CONVERT input for this component. Ten records were used to completely describe the component.

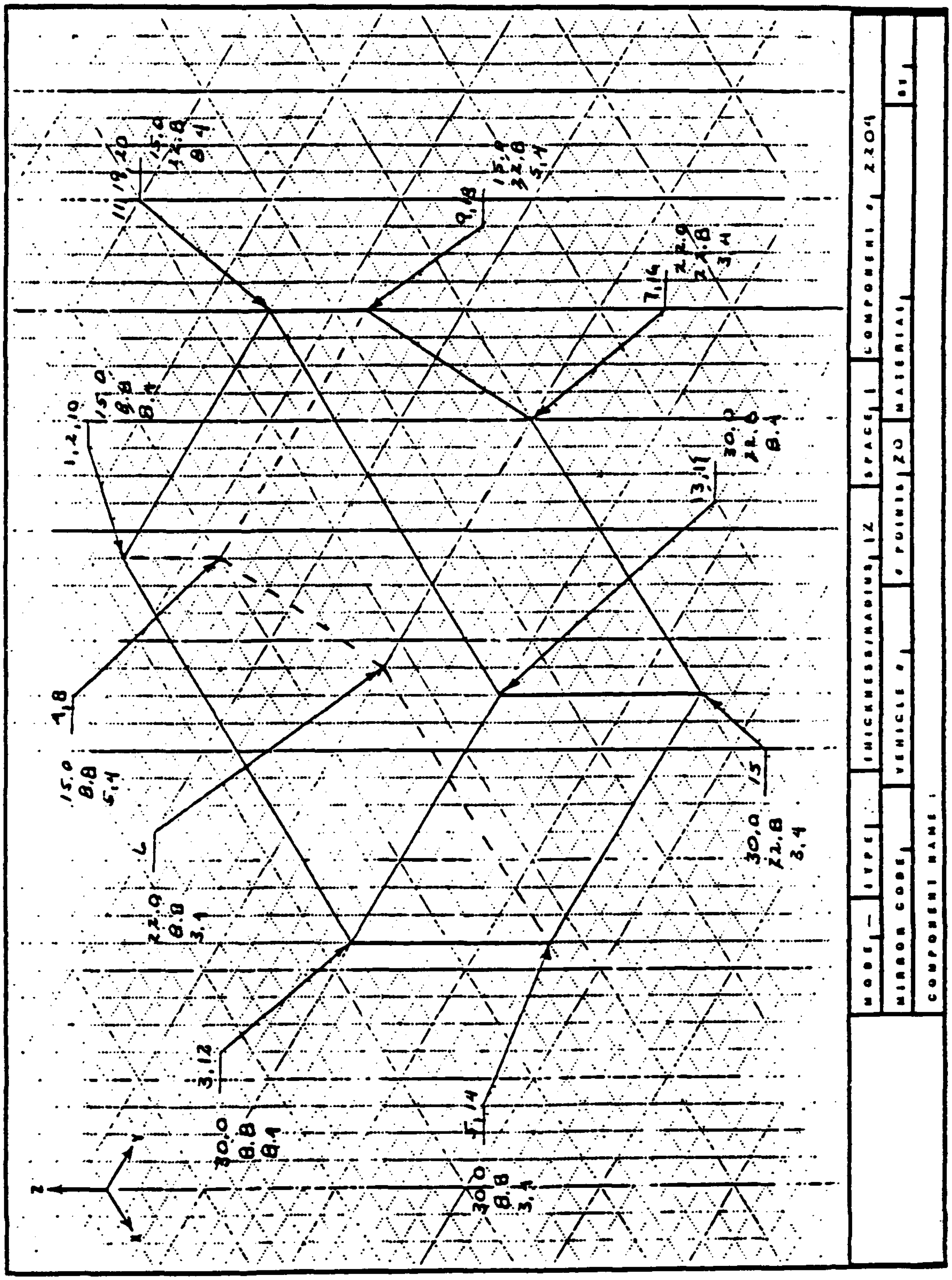


Figure 17. Sketch of Flat-Sided Object

X COORD OR RADIUS	Y COORD OR RADIUS	Z COORD	±GNNS	COMP CODE	SEQUENCE NUMBER	ADDITIONAL SEQUENCE NUMBERS							MIRROR	VC
00000000 12345678	01111111 90123456	111222222 789012345	222233 678901	3333 2345	33334444444 67890123456	4445 7890	5555 1234	5555 5678	5666 9012	6666 3456	6667 7890	7777 1234	777 567	778 890
15.00	8.80	8.40	-121	2204	10	20	100							999
30.00	8.80	8.40	-121	2204	30	120								999
15.00	8.80	5.40	-121	2204	40	80								999
30.00	8.80	3.40	-121	2204	50	140								999
22.00	8.80	3.40	-121	2204	60									999
22.00	22.80	3.40	-121	2204	70	160								999
15.00	22.80	5.40	-121	2204	90	180								999
15.00	22.80	8.40	-121	2204	110	190	200							999
30.00	22.80	8.40	-121	2204	130	170								999
30.00	22.80	3.40	-121	2204	150									999

Figure 18. Coding for Flat-Sided Component

The first three entries on each card contain the X, Y, and Z coordinates of each point. The next entry contains the descriptive code followed by the component code number. The next eight entries are reserved for sequence numbers. The modeler may enter multiple sequence numbers as shown in Figure 18, or elect to create separate records for each sequence point. Notice that the sequence numbers are entered in multiples of 10, i.e., 10, 20, 30, 40, etc., rather than as 1, 2, 3, 4, etc. Defining sequence numbers in this manner allows the addition of intermediate sequence numbers, if needed, without having to redefine all succeeding sequence numbers. Notice also that leading zeros in the description codes may be omitted.

An example of a more complex component modeled using triangle primitives is illustrated in Figure 19. The component was modeled in plate mode. The top curved surface was modeled first, ending with point 12, then continued to describe the back surface through point 22. In doing so, a triangle was defined on the bottom surface (points 11, 12 and 13). Continuing from point 22, the remaining area of the bottom surface was described ending at point 31. The modeler then created a degenerate triangle (points 30, 31 and 32) and continued describing the front surface ending with a repeated point (points 40 and 41). The sequencing of points, while perhaps not the most efficient, is perfectly legitimate in that the entire surface of the component was described only one time. Figure 20 shows the data description input to CONVERT.

b. Box

If rectangular parallelepipeds (or boxes) are used to approximate a component, each surface can be defined with adjacent triangle primitives formed by a minimum of 16 sixteen sequenced points as shown in Figure 21(a). As an alternative, CONVERT permits describing a box by defining the coordinates of only four of the vertices or corners of the box. This is accomplished by first defining the X, Y, Z coordinates of one corner on the first record, and then defining the coordinates of the three corners adjacent to the first corner on the next three records, as illustrated in Figure 21(b). CONVERT will convert the box using triangle primitives and 16 points.

The box can be described in either volume or plate mode. The form of the description code for the box shown in Figure 21(b) is ±7NNS where the plus(+) or minus(-) specifies volume or plate mode respectively, the 7 indicates that the primitive is a box, NN is the normal thickness (in inches x 100 if plate mode, or 00 if volume mode), and the S specifies the space code. An illustration of a component described using three box approximations is shown in Figure 22 and coded in Figure 23. Each box is modeled in volume mode (700S)

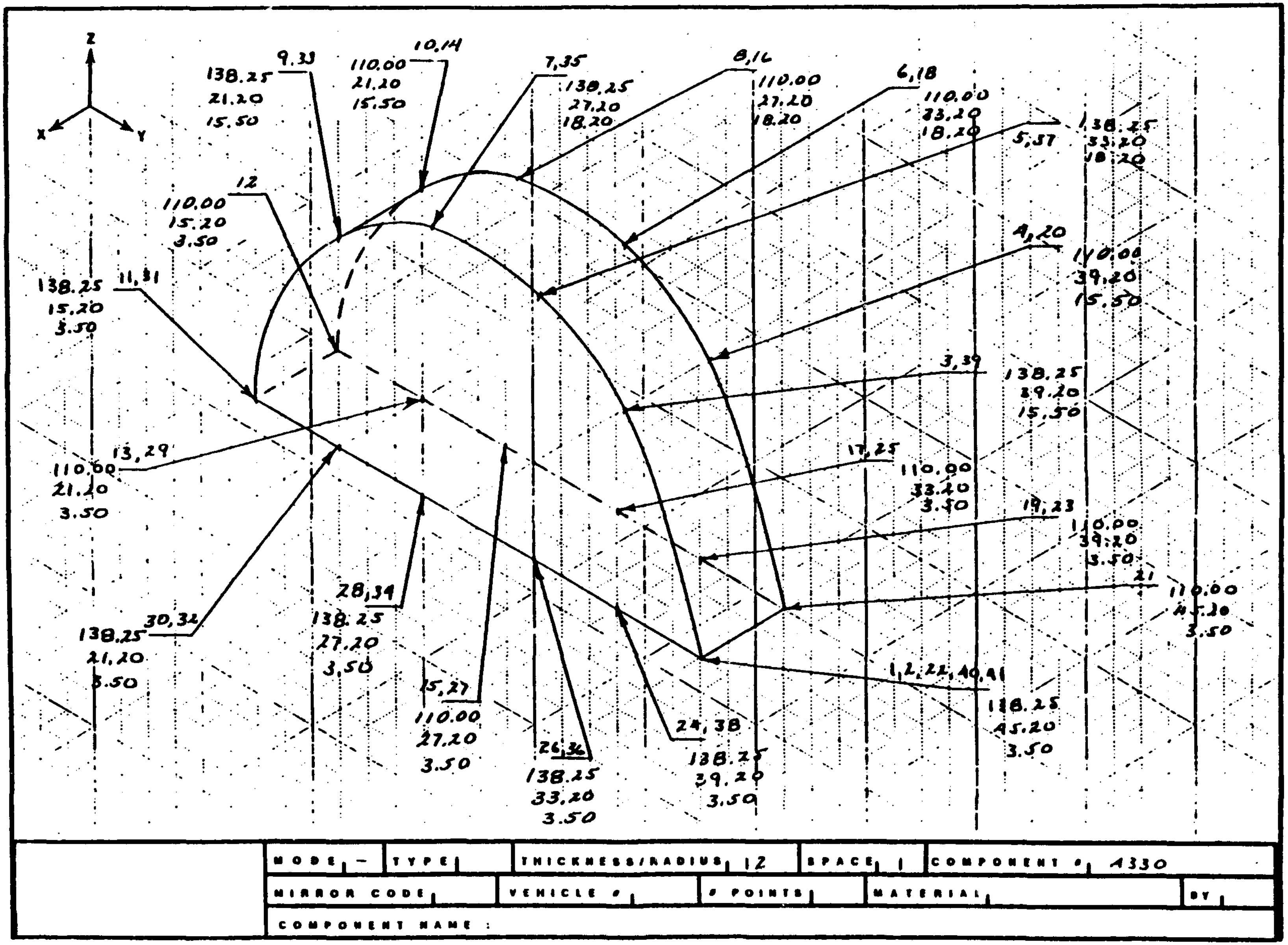
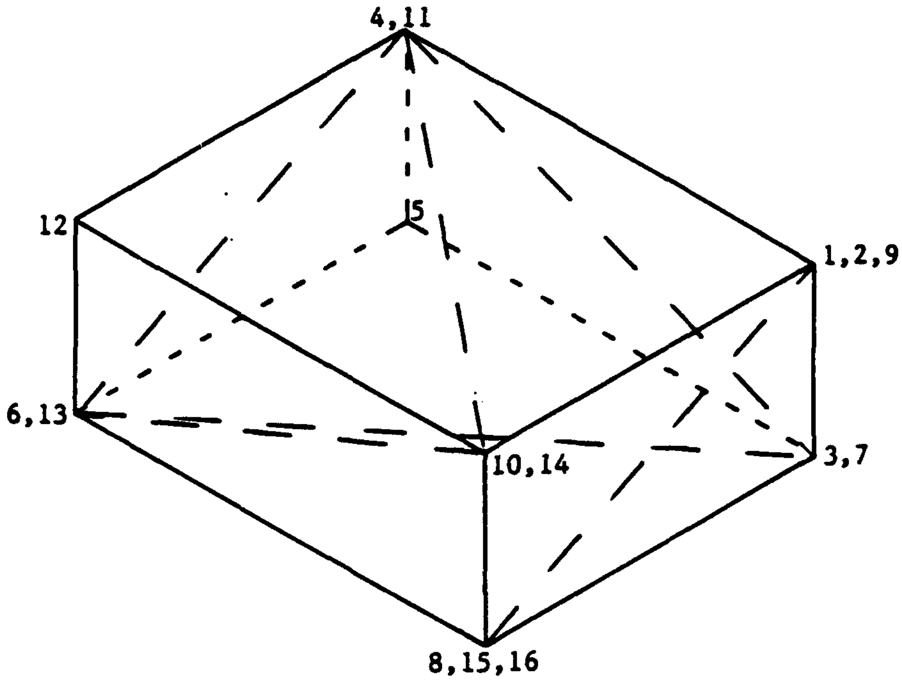


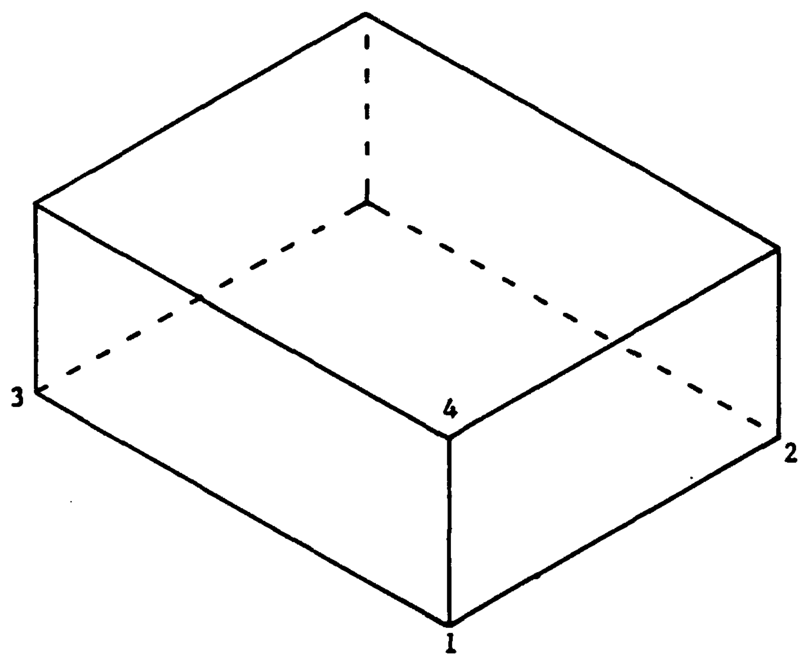
Figure 19. Sketch of Complex Component

X COORD OR RADIUS	Y COORD OR RADIUS	Z COORD	±GNNS	COMP CODE	SEQUENCE NUMBER	ADDITIONAL SEQUENCE NUMBERS							MIRROR	VC
00000000 12345678	01111111 90123456	111222222 789012345	222233 678901	3333 2345	33334444444 67890123456	4445 7890	5555 1234	5555 5678	5666 9012	6666 3456	6667 7890	7777 1234	777 567	778 890
138.25	45.20	3.50	-121	4330	10	20	220	400	410					999
138.25	39.20	15.50	-121	4330	30	390								999
110.00	39.20	15.50	-121	4330	40	200								999
138.25	33.20	18.20	-121	4330	50	370								999
110.00	33.20	18.20	-121	4330	60	180								999
138.25	27.20	18.20	-121	4330	70	350								999
110.00	27.20	18.20	-121	4330	80	160								999
138.25	21.20	15.50	-121	4330	90	330								999
110.00	21.20	15.50	-121	4330	100	140								999
138.25	15.20	3.50	-121	4330	110	310								999
110.00	15.20	3.50	-121	4330	120									999
110.00	21.20	3.50	-121	4330	130	290								999
110.00	27.20	3.50	-121	4330	150	270								999
110.00	33.20	3.50	-121	4330	170	250								999
110.00	39.20	3.50	-121	4330	190	230								999
110.00	45.20	3.50	-121	4330	210									999
138.25	39.20	3.50	-121	4330	240	380								999
138.25	33.20	3.50	-121	4330	260	360								999
138.25	27.20	3.50	-121	4330	280	340								999
138.25	21.20	3.50	-121	4330	300	320								999

Figure 20. Coding For Complex Components



(a)



(b)

Figure 21. Box Approximation Using: (a) 16 Points and (b) 4 Points

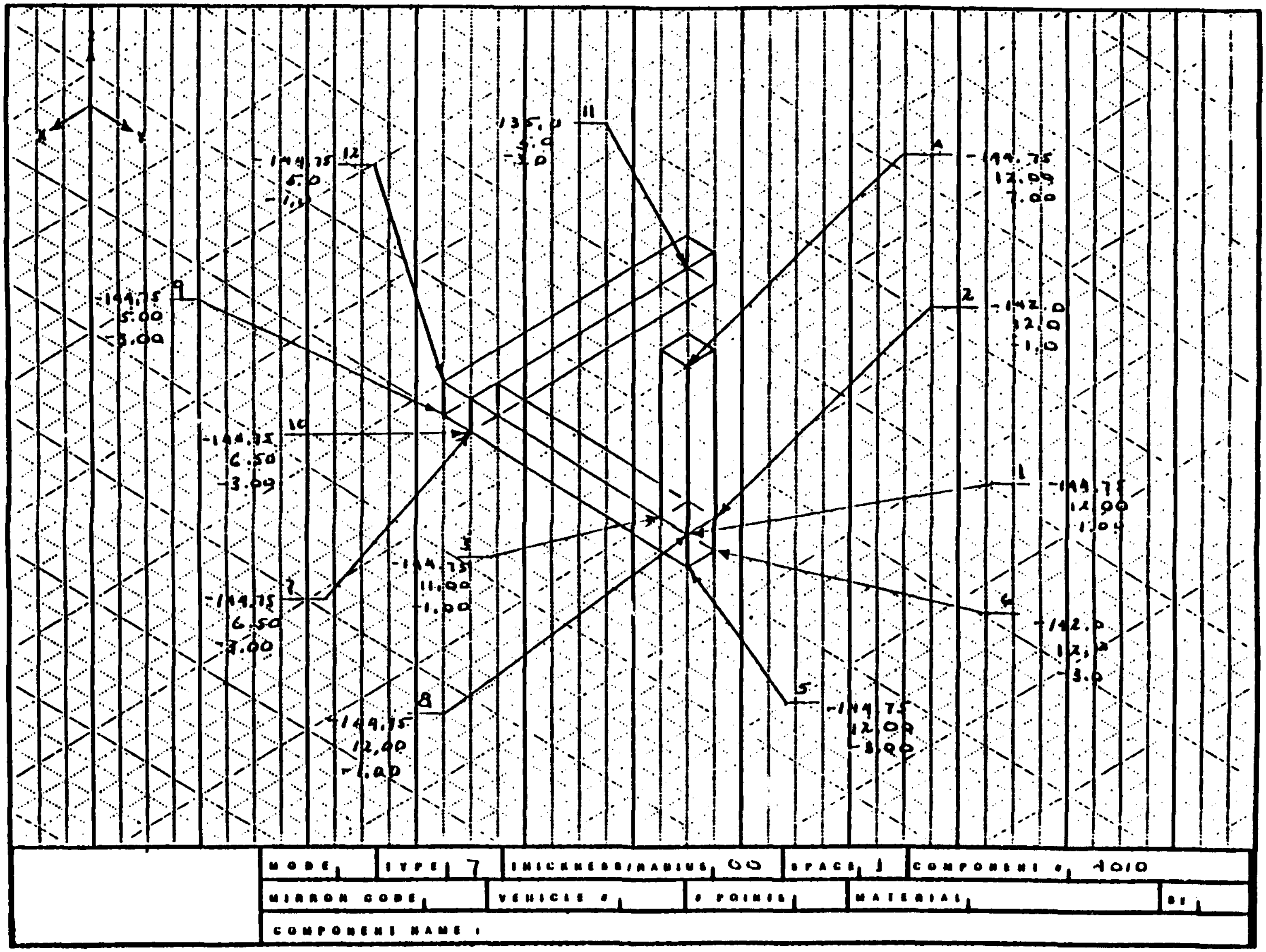


Figure 22. Sketch of Component Described With Three Boxes

X COORD OR RADIUS	Y COORD OR RADIUS	Z COORD	±GHNS	COMP CODE	SEQUENCE NUMBER	ADDITIONAL SEQUENCE NUMBERS							MIRROR	VC
00000000 12345678	01111111 90123456	111222222 789012345	222233 678901	3333 2345	33334444444 67890123456	4445 7890	5555 1234	5555 5678	5666 9012	6666 3456	6667 7890	7777 1234	777 567	778 890
-144.75	12.00	-1.00	7001	4010	10									999
-142.00	12.00	-1.00	7001	4010	20									999
-144.75	11.00	-1.00	7001	4010	30									999
-144.75	12.00	7.00	7001	4010	40									999
-144.75	12.00	-3.00	7001	4010	50									999
-142.00	12.00	-3.00	7001	4010	60									999
-144.75	6.50	-3.00	7001	4010	70									999
-144.75	12.00	-1.00	7001	4010	80									999
-144.75	5.00	-3.00	7001	4010	90									999
-144.75	6.50	-3.00	7001	4010	100									999
-135.00	5.00	-3.00	7001	4010	110									999
-144.75	5.00	-1.00	7001	4010	120									999

Figure 23. Coding for Component Described With Three Boxes

using four points. The modeler chose sequence numbers 10 through 40 for one box, 50 through 80 for the second box, and 90 through 120 for the remaining box.

c. Wedge

Right angle wedges can be defined with triangle primitives using a minimum of 12 sequence points as shown in Figure 24(a) or by a simplified method of defining four points as shown in Figure 24(b). Referring to Figure 24(b), Points 1, 2, and 4 define the vertices of one of the triangular faces of the wedge while Points 2 and 4 define the hypotenuse of the triangle. Point 3 defines the other vertex which is adjacent to Point 1. The inputs required to describe a wedge consist of four records consisting of the X, Y, Z coordinates of the four vertices along with the description code and component code number.

The wedge can be described in either the volume or plate mode. The form of the description code for a wedge is ±5NNS where the + or - specifies volume or plate mode respectively, the 5 indicates that the primitive is a wedge, NN is the normal thickness (in inches x 100 if plate mode, or 00 if volume mode), and the S specifies the space code. An illustration of a component described using box and wedge primitives is shown in Figure 25 and coded in Figure 26. The entire component was modeled in plate mode with a plate thickness of 0.12-inch.

d. Cylinders and Cones

If a cylinder primitive is used to approximate a component, only four dimensions on three records are required to define the surface. These are the coordinates of the two end points and the magnitude of the two radii (measured at the end points) as illustrated in Figure 27(a) and (b). If the two radii are equal in magnitude, a cylinder is described. If the radii are unequal, a truncated cone is described. When coding a cylinder, enter the X, Y, and Z coordinates of the two end points on the first two records, then enter the radii corresponding to the first and second record coordinates in the X- and Y-fields respectively, of the third record. A code which specifies whether one or both of the ends of the cylinder are open or closed is entered in the Z-field of the third card. These codes are defined in Table 6.

Cylinder primitives can be used to describe components in either the volume or plate mode. Cylinders are described with a descriptive code in the form of ±4NNS or ±8NNS where the + or - specifies volume or plate mode respectively, NN is the normal thickness (in inches x 100 if plate mode, or 00 if volume mode), and the S specifies the space

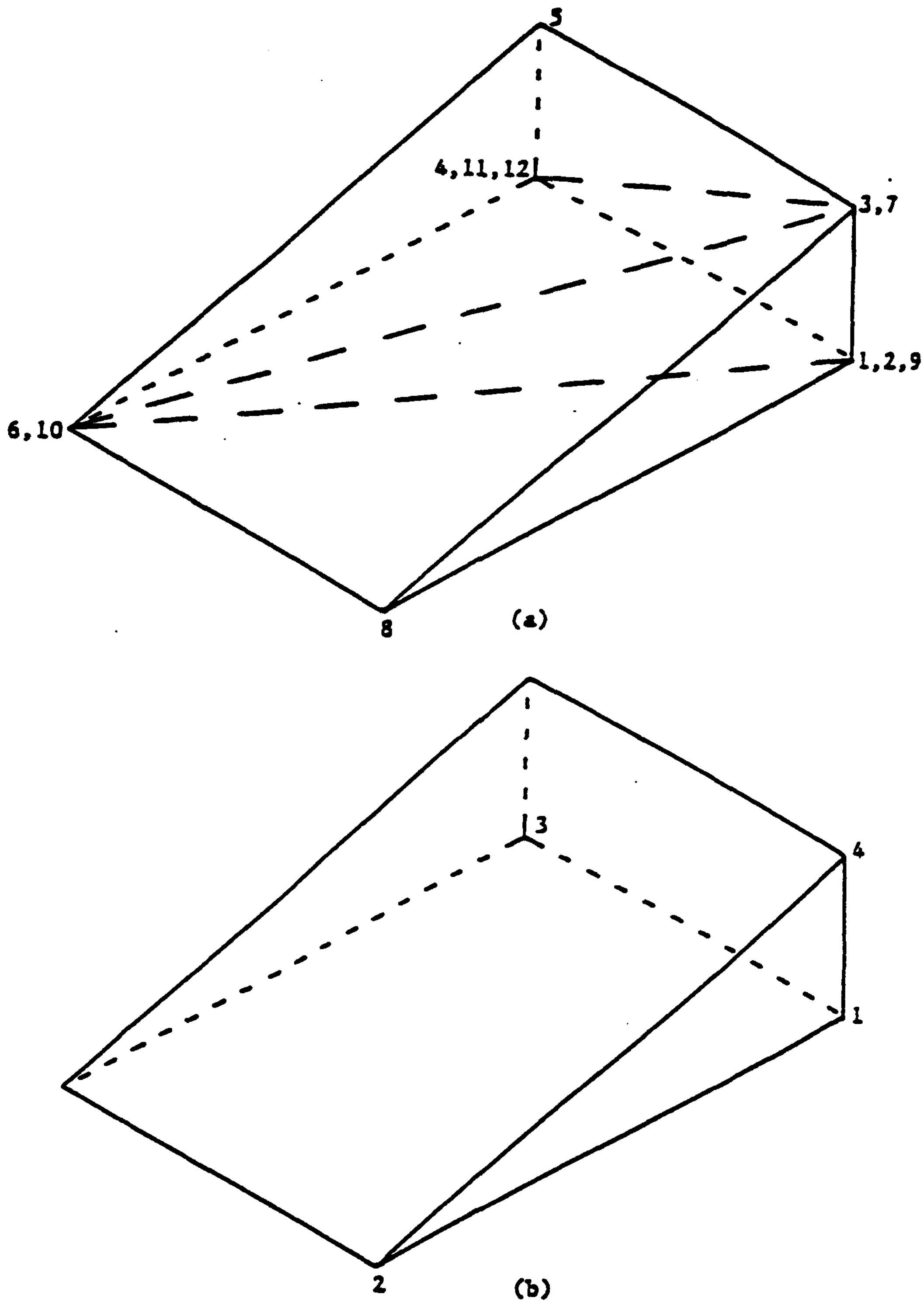


Figure 24. Wedge Approximation Using: (a) 12 Points and (b) 4 Points

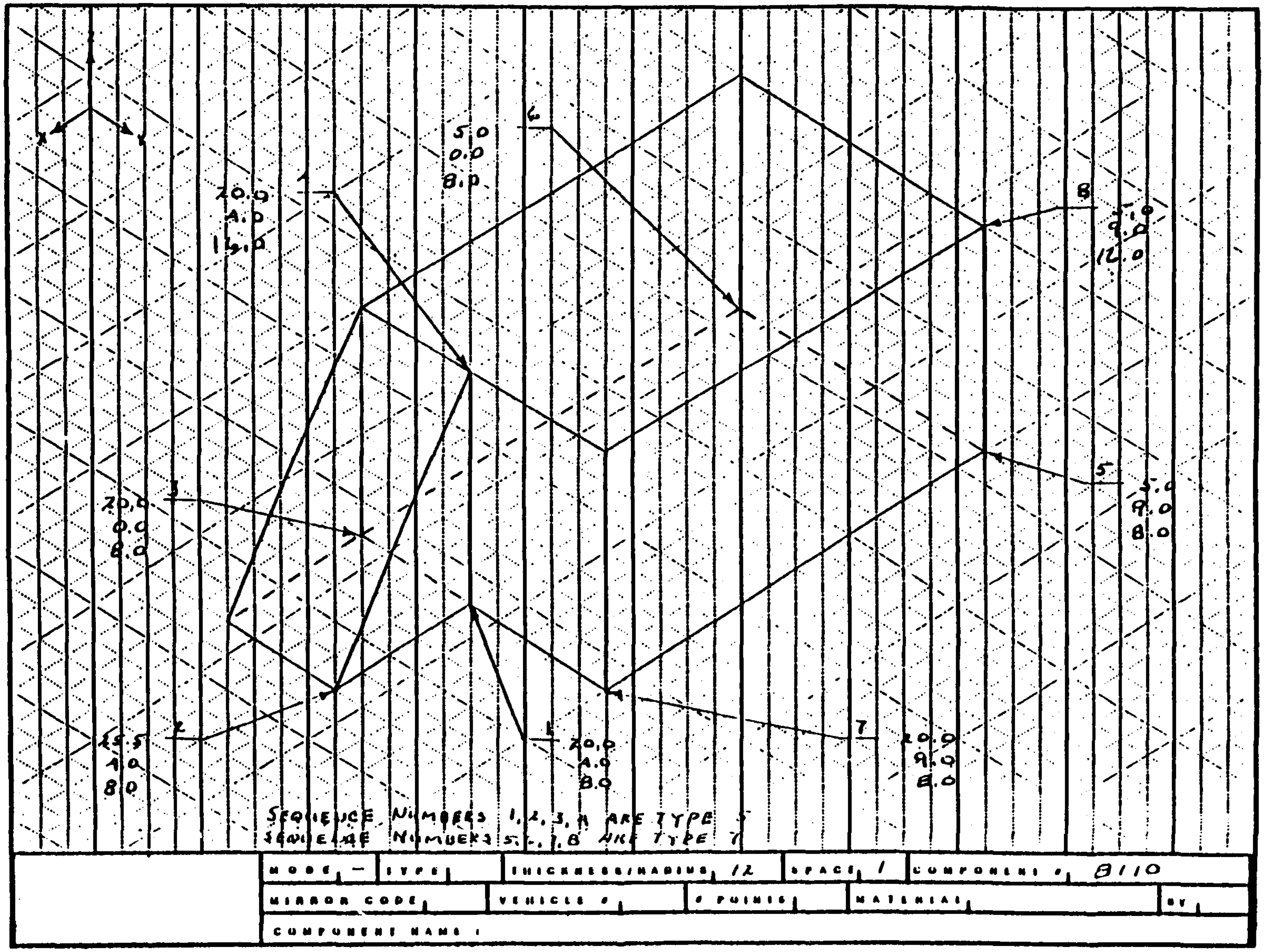


Figure 25. Sketch of a Component Modeled with Box and Wedge Primitives

X COORD OR RADIUS	Y COORD OR RADIUS	Z COORD	±GNNS	COMP CODE	SEQUENCE NUMBER	ADDITIONAL SEQUENCE NUMBERS							MIRROR	VC
00000000 12345678	01111111 90123456	111222222 789012345	222233 678901	3333 2345	33334444444 67890123456	4445 7890	5555 1234	5555 5678	5666 9012	6666 3456	6667 7890	7777 1234	777 567	778 890
20.00	4.00	8.00	-5121	8110	10									999
25.50	4.00	8.00	-5121	8110	20									999
20.00	0.00	8.00	-5121	8110	30									999
20.00	4.00	16.00	-5121	8110	40									999
5.00	9.00	8.00	-7121	8110	50									999
5.00	0.00	8.00	-7121	8110	60									999
20.00	9.00	8.00	-7121	8110	70									999
5.00	9.00	16.00	-7121	8110	80									999

Figure 26. Coding for a Wedge/Box Component

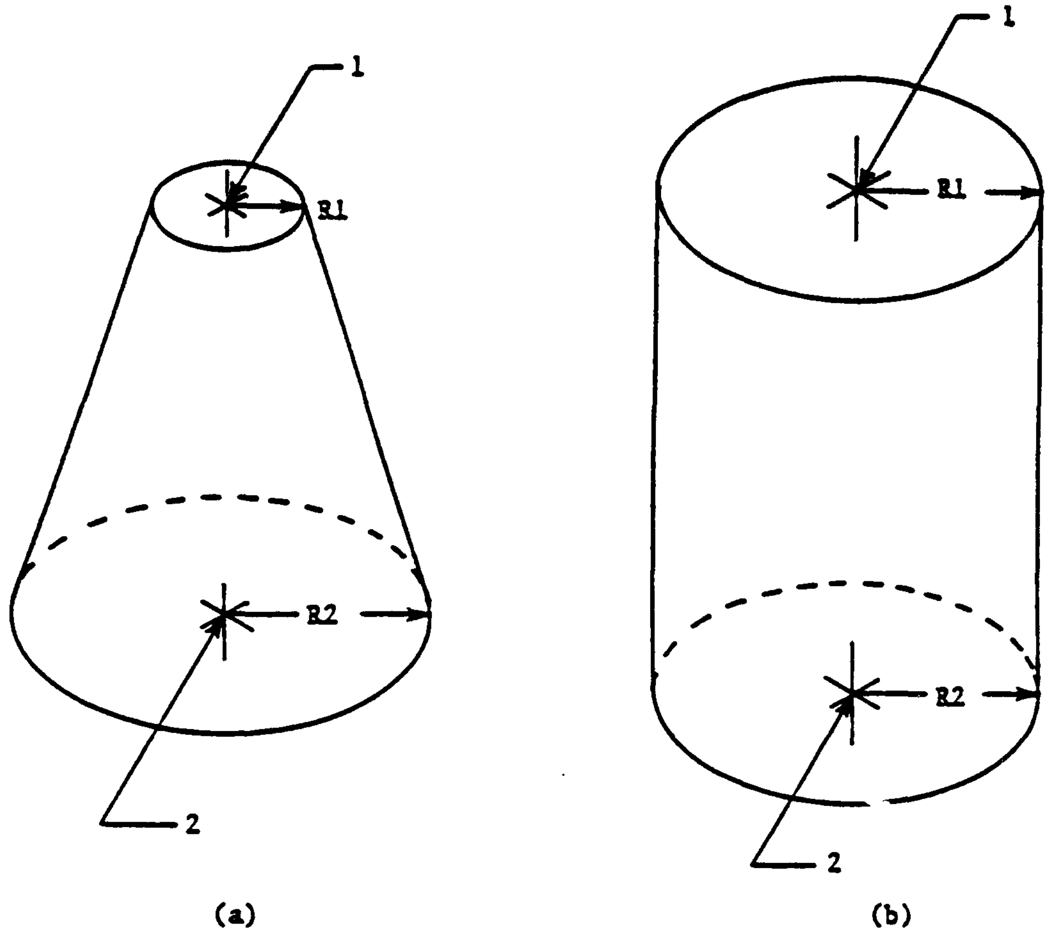


Figure 27. Simplified Method of Describing Cones and Cylinders

TABLE 6. CODES INDICATING CONDITION OF CONES AND CYLINDER ENDS

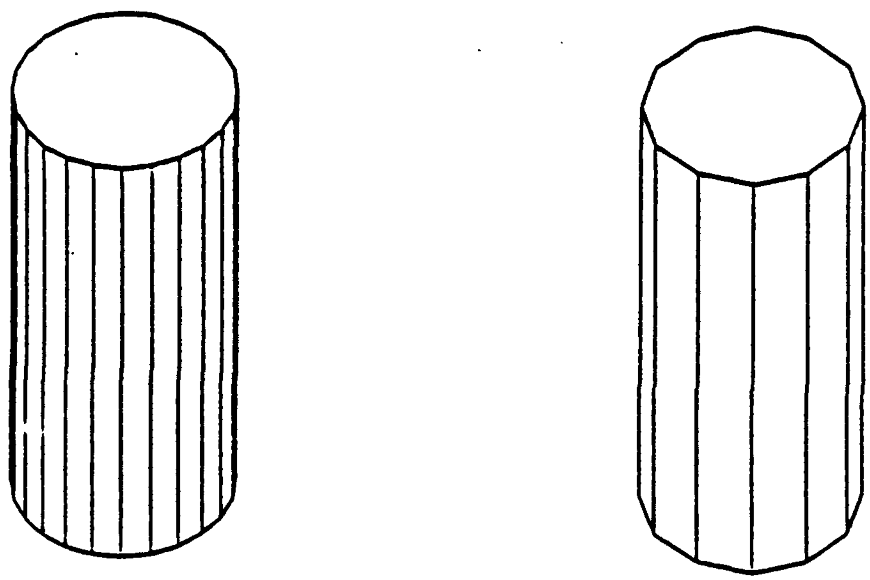
CODE NUMBER	MODE	END DEFINED BY FIRST RECORD	END DEFINED BY SECOND RECORD
0	PLATE VOLUME	OPEN CLOSED	OPEN CLOSED
1	PLATE VOLUME	CLOSED CLOSED	OPEN OPEN
2	PLATE VOLUME	OPEN OPEN	CLOSED CLOSED
3	PLATE VOLUME	CLOSED OPEN	CLOSED OPEN

code. The G of the description code has two purposes. First, either a 4 or an 8 assigned to G defines the primitive as a cylinder. The second purpose applies only if the output file of CONVERT is to be used for plotting and cylinders are to be transformed (i.e., the IV parameter on the CONVERT Type 1 Run Specification Card is set equal to 1). Then, if the G is specified as a 4, the cone or cylinder will be transformed using 24 points equally spaced around the outer edge of each end as illustrated in Figure 28(a). If the G is specified as an 8, the cone or cylinder will be transformed using 12 points as illustrated in Figure 28(b). If so desired, the user may flag individual cylinders to be transformed to 4, 6, 8, 12, 24, or 32 point cylinders. This is accomplished by inserting the appropriate value (4, 6, 8, 12, 24, or 32) in the third sequence point location (columns 51 through 54) of each of the three input records describing a cylinder.

Figure 29 illustrates a component formed by two cones and one cylinder. Each primitive of the component is described in plate mode with a normal thickness of 0.12 inches. Notice, in the coding shown in Figure 30, that the larger radii of both truncated cones are coded "closed" and the smaller radii of both are coded "open" by placing 1.00 in the Z-field of the third record and 2.00 in the Z-field of the last record. By placing 0.00 in the Z-field of the sixth record, both ends of the cylinder are coded as "open" thus forming a sealed tank of sorts. In plate mode, the end covers will have the same normal thickness as the cylinder primitives 0.12-inch). Also notice the 6 entered in the third sequence number position (columns 52-55). Without this entry, the cones and cylinder, if the option were selected, would be transformed with 12 points (the G of GNNS is equal to 8). By entering the 6, the transformation using 12 points is overridden and each primitive will be transformed with 6 points equally spaced around their ends.

e. Spheres

When a sphere primitive is used to approximate a component, only the center point and the radius of the sphere are required to define the entire surface of the sphere as illustrated in Figure 31. Two records are required to describe a sphere. The first record defines the X, Y, and Z coordinates of the sphere center. The radius of the sphere is entered in the X-field of the second record. Enter zeros in the Y- and Z-fields of the second record.



(a) (b)

Figure 28. (a) 24-Point Cylinder and (b) 12-Point Cylinder

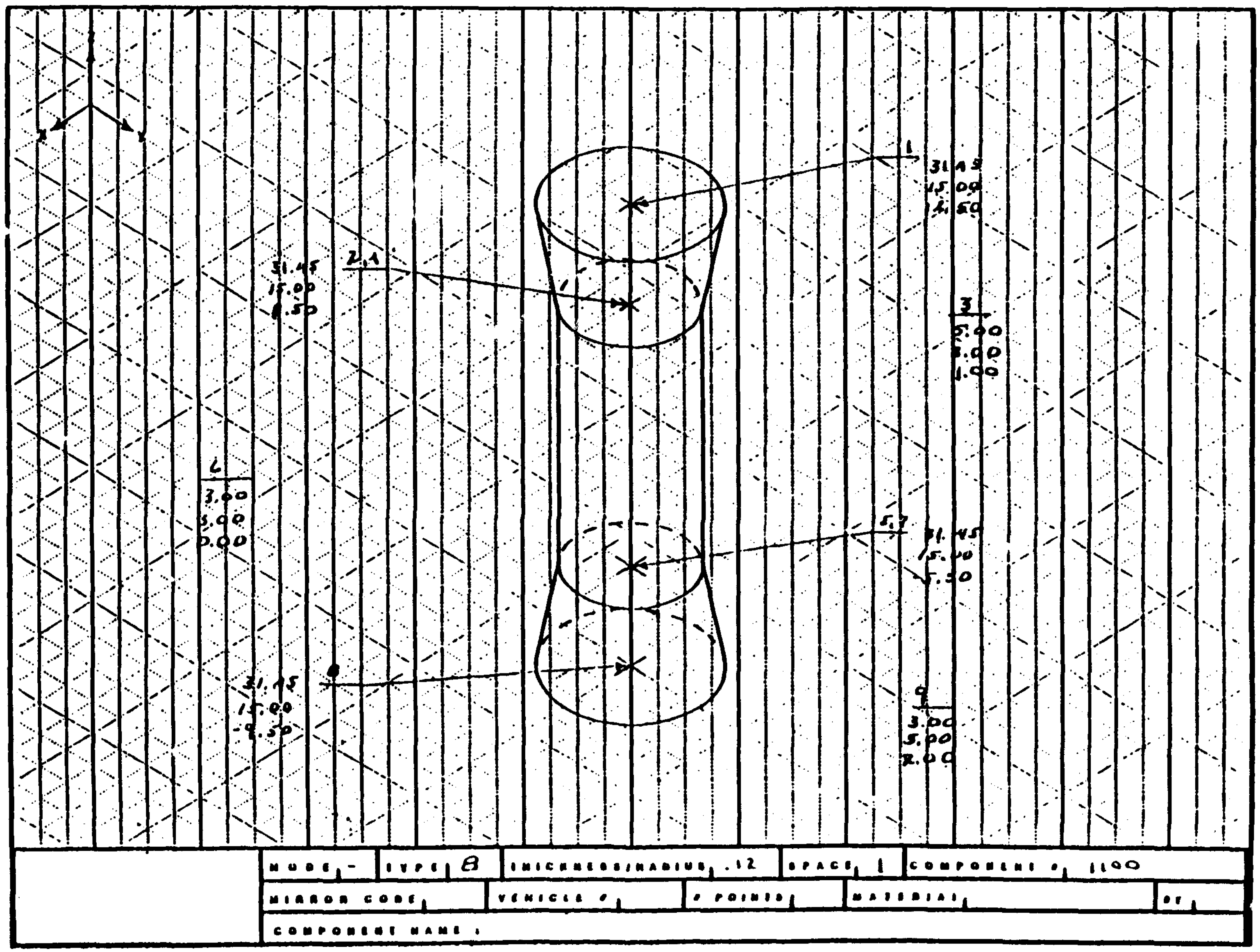


Figure 29. Sketch of Sealed Tank Formed by Two Truncated Cones and a Cylinder

X COORD OR RADIUS	Y COORD OR RADIUS	Z COORD	±GNNS	COMP CODE	SEQUENCE NUMBER	ADDITIONAL SEQUENCE NUMBERS							MIRROR	VC
00000000 12345678	01111111 90123456	111222222 789012345	222233 678901	3333 2345	33334444444 67890123456	4445 7890	5555 1234	5555 5678	5666 9012	6666 3456	6667 7890	7777 1234	777 567	778 890
31.45	15.00	12.50	-8121	1100	10		6							999
31.45	15.00	8.50	-8121	1100	20		6							999
5.00	3.00	1.00	-8121	1100	30		6							999
31.45	15.00	8.50	-8121	1100	40		6							999
31.45	15.00	-5.50	-8121	1100	50		6							999
3.00	3.00	0.00	-8121	1100	60		6							999
31.45	15.00	-5.50	-8121	1100	70		6							999
31.45	15.00	-9.50	-8121	1100	80		6							999
3.00	5.00	2.00	-8121	1100	90		6							999

Figure 30. Coding for Sealed Tank Formed by Two Truncated Cones and a Cylinder

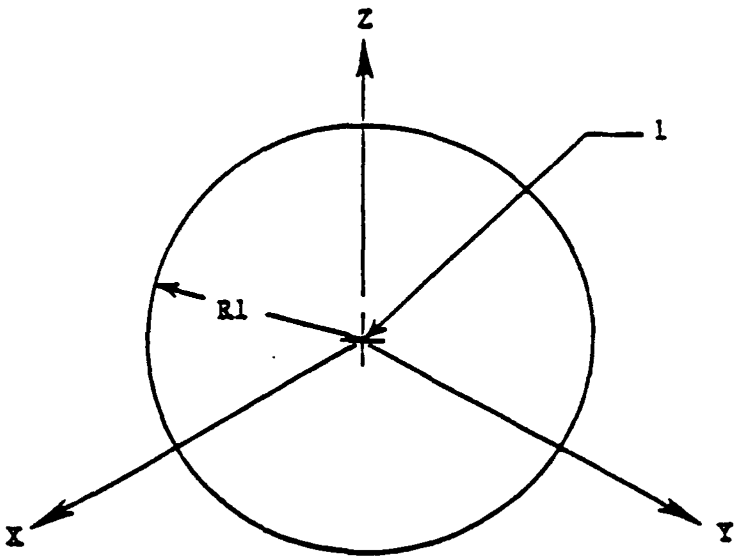


Figure 31. Simplified Method of Describing Spheres

A sphere primitive may be used to approximate a component in either the plate or volume mode. The form of the description code for a sphere is ±6NNS where the + or - specifies volume or plate mode respectively, the 6 indicates a sphere, the NN specifies normal thickness (in inches x 100 if plate mode, or 00 if volume mode), and the S specifies the space code. Figure 32 illustrates a spherical component described in the plate mode. The normal thickness of the sphere shell is 0.25-inch. Figure 33 shows the descriptive coding for this component.

f. Rod Mode

The rod mode primitive is useful when modeling small diameter cylindrical components. Such components include fuel lines, oil lines, control rods and cables, and electrical wiring. Rod mode components are specified by defining the coordinates of the end points and radius for each straight line segment of the rod. The description code for a rod mode component is in the form of -9NNS where -9 specifies rod mode, NN specifies the radius (in inches x 100), and S specifies the space code. Since the largest value that can be inserted into the NN position is 99, the diameter of a rod mode component is limited to a maximum of 1.98-inches.

There are several rules and cautions associated with modeling rod mode components which must be observed. These are described in detail in the paragraphs that follow.

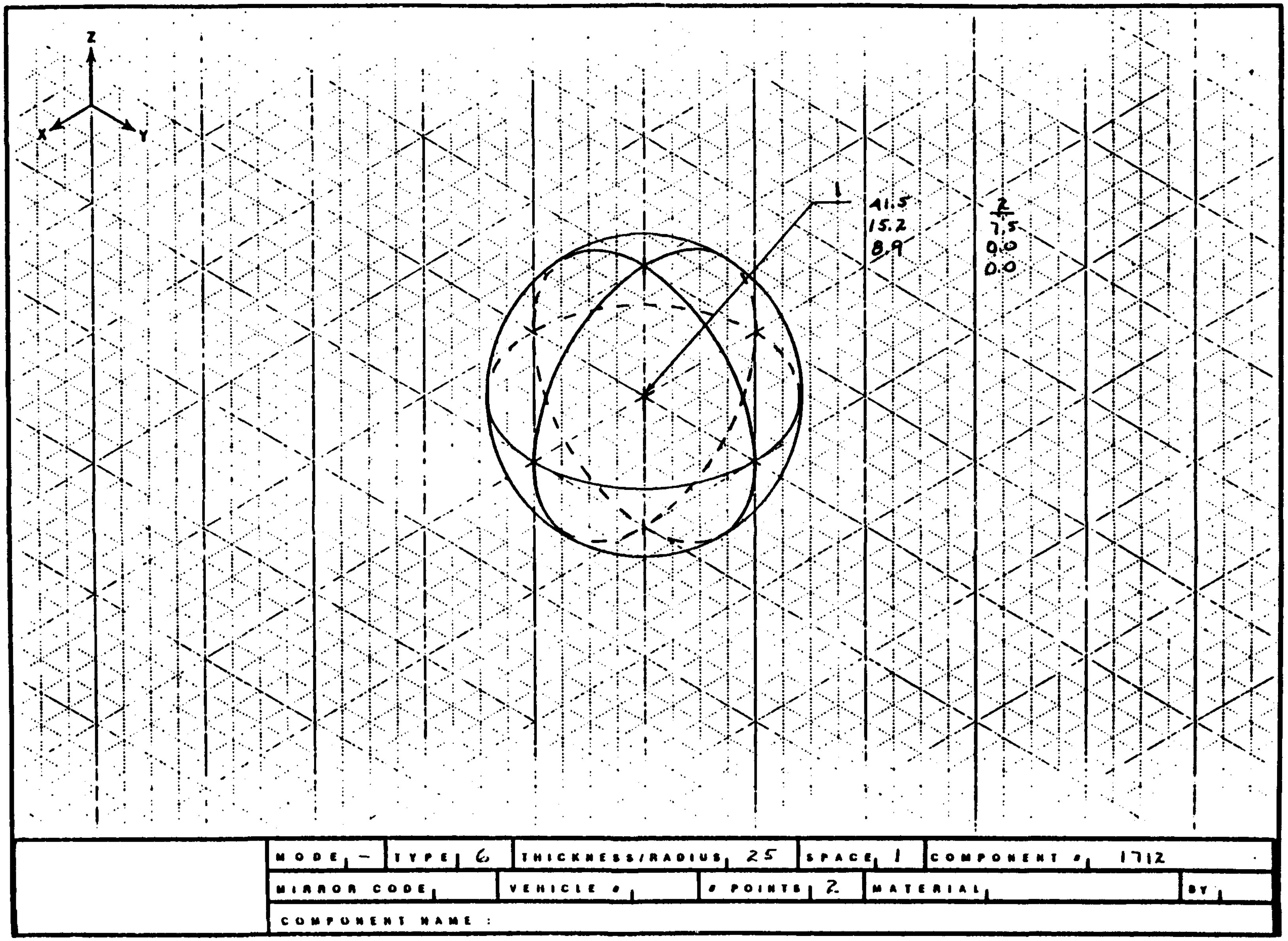


Figure 32. Sketch of Spherical Component

X COORD OR RADIUS	Y COORD OR RADIUS	Z COORD	±GNNS	COMP CODE	SEQUENCE NUMBER	ADDITIONAL SEQUENCE NUMBERS							MIRROR	VC
00000000 12345678	01111111 90123456	111222222 789012345	222233 678901	3333 2345	33334444444 67890123456	4445 7890	5555 1234	5555 5678	5666 9012	6666 3456	6667 7890	7777 1234	777 567	778 890
41.50	15.20	8.90	-6251	1712	10									999
7.50	0.00	0.00	-6251	1712	20									999

Figure 33. Coding for a Spherical Component

(1) **Rod Mode Components With Multiple Ends.** Each component modeled in rod mode should have not more than two of its ends which connect to other components. For example, one could properly model in rod mode, the whip antenna on a vehicle (a connection on one end), or model a wire running from the battery to the starter of a truck (a connection on two ends). In the later case, one should not model a segment of wire as being connected to the wire between the battery and the starter and route it to a third component, and then define the entire wire pack as a single component. A break in the wire could be interpreted, in vulnerability programs, as isolating the starter from the battery, isolating the third component from the battery, or both. Even giving the added wire a component name different from that given to the wire routed from the battery to starter does not solve the problem as a break in the battery to starter wire could be interpreted as isolating the starter, or isolating both the starter and the third component. The proper modeling technique would define the wires as three components; one from the starter to a junction, one from the battery to the junction, and one from the third component to a junction.

(2) **Sequence Numbers.** When rod mode components are coded, they are given repeated sequence numbers for both the first and last points and for those points where a discontinuity occurs. Single sequence numbers are given to points where the routing of a component is altered. A discontinuity might occur where the radius of the rod mode component is changed, a change is space code occurs, or a break in the line is modeled. Figure 34 illustrates the sequencing of points for a rod mode component. Sequence points 1 and 2 are repeated points for one end of the component and points 3 and 4 are single points defining a point where routing is altered. A break in the rod between points 5,6 and 7,8 would be modeled if the component were to pass through some form of shielding such as a bulkhead. Points 9 and 10 define a change in component routing and points 11 and 12 are repeated points defining the end of the rod. In examining Figure 34, one should note that the sketch and sequencing could represent the sequencing of one component with a change in radius and/or a change in space code assigned to the segment after the break since the break is represented by repeated points.

(3) **Long Rods.** A potential problem with a rod mode component is that they may be modeled as a long and sometimes complicated object having one component name. This is a poor modeling practice as such a component is difficult to debug and inefficient to process. The modeler is advised that computer time can be saved and modeling errors reduced if rod mode components are broken into short segments and given different component code numbers.

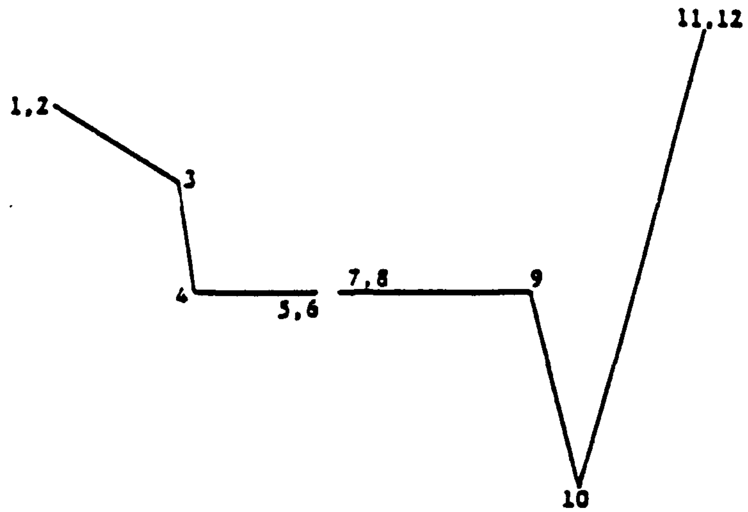


Figure 34. Point Sequencing of a Rod Mod Component

(4) **Threat Radius.** The advantage of using rod mode primitives instead of cylinder primitives is that the effect of projectile diameter and its interaction with a component can be considered when executing FASTGEN. For example, when a rod mode component is modeled, it is defined by a minimum of two end points and a radius. That radius is applied to the whole length of the component including the end points. FASTGEN allows the input of a second radius called a threat radius to be added to the radius of rod mode components to account for the presented area of fragments or of the projectile being considered. This can create problems because the radius of the threat is considered with the rod mode primitives but not with any other primitives. Figure 35(a) shows an intercept registered for a rod component located just inside the aircraft skin, but no intercept is registered for the skin. The problem is illustrated better when redrawn as in Figure 35(b). Allowing both the rod and the interfering component intercept on the ray would overstate the ballistic resistance of the shotline. A way to avoid this error is to position the rod, allowing for both the modeled radius and the threat radius, so that the rod is not too close to other components.

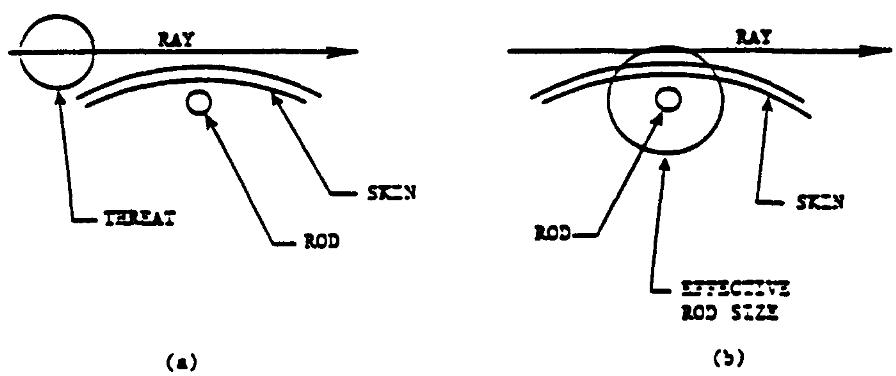


Figure 35. Rod/Threat Interaction

Figure 36 shows a sketch of two wires (labeled A and B) connected to a third wire (labeled C). All three components are modeled in rod mode. Figure 37 illustrates coding for these three components.

g. Donuts

The donut primitive is composed of two cylinders, one inside the other as illustrated in Figure 38. (The axes of the cylinders do not have to coinside.) There are several options available for enclosing the ends between the outer edge of the inner cylinder and the inner edge of the outer cylinder, but all options leave both ends of the inner cylinder open (hence the name donut).

Donuts are described with a descriptive code in the form of ±3NNS where the + or - specifies volume or plate mode, respectively, NN is the radius (in inches x 100 if plate mode, or 00 if volume mode) and the S specifies the space code. Six records are required to describe a donut primitive. The first three records pertain to the outer cylinder and the second three records pertain to the inner cylinder. If one thinks of the cylinders as having ends A and B, then data are entered into the X-, Y-, and Z-fields of these records as follows:

Record 1	Coordinates of end A of the outer cylinder.
Record 2	Coordinates of end B of the outer cylinder.
Record 3	Radius of the outer cylinder at end A in the X-field.
	Radius of the outer cylinder at end B in the Y-field.
Record 4	Coordinates of end A of the inner cylinder.
Record 5	Coordinates of end B of the inner cylinder.
Record 6	Radius of the inner cylinder at end A in the X-field.
	Radius of the inner cylinder at end B in the Y-field.

If the donut primitive is described in volume mode the modeler should enter only code 0 in the Z-field of record 6. This will close the volume that exists between the inner and outer cylinders on both A and B ends. If the donut primitive is described in plate mode, there are seven options available as defined in Table 7 and illustrated in Figure 39. The code for the option selected must be entered in the Z-field of record 6. In all cases, the Z-field of record 3 should contain a zero. Figure 40 shows two end views of four components modeled in volume mode using donut primitives. Figure 41 illustrates the coding for the four components.

If donuts are to be plotted, the default number of points around the ends is 12 as shown in Figures 39 and 40. This may be overridden by entering in one of the codes described in Table 7 into the third sequence number position (columns 52-55) of each of the six records.

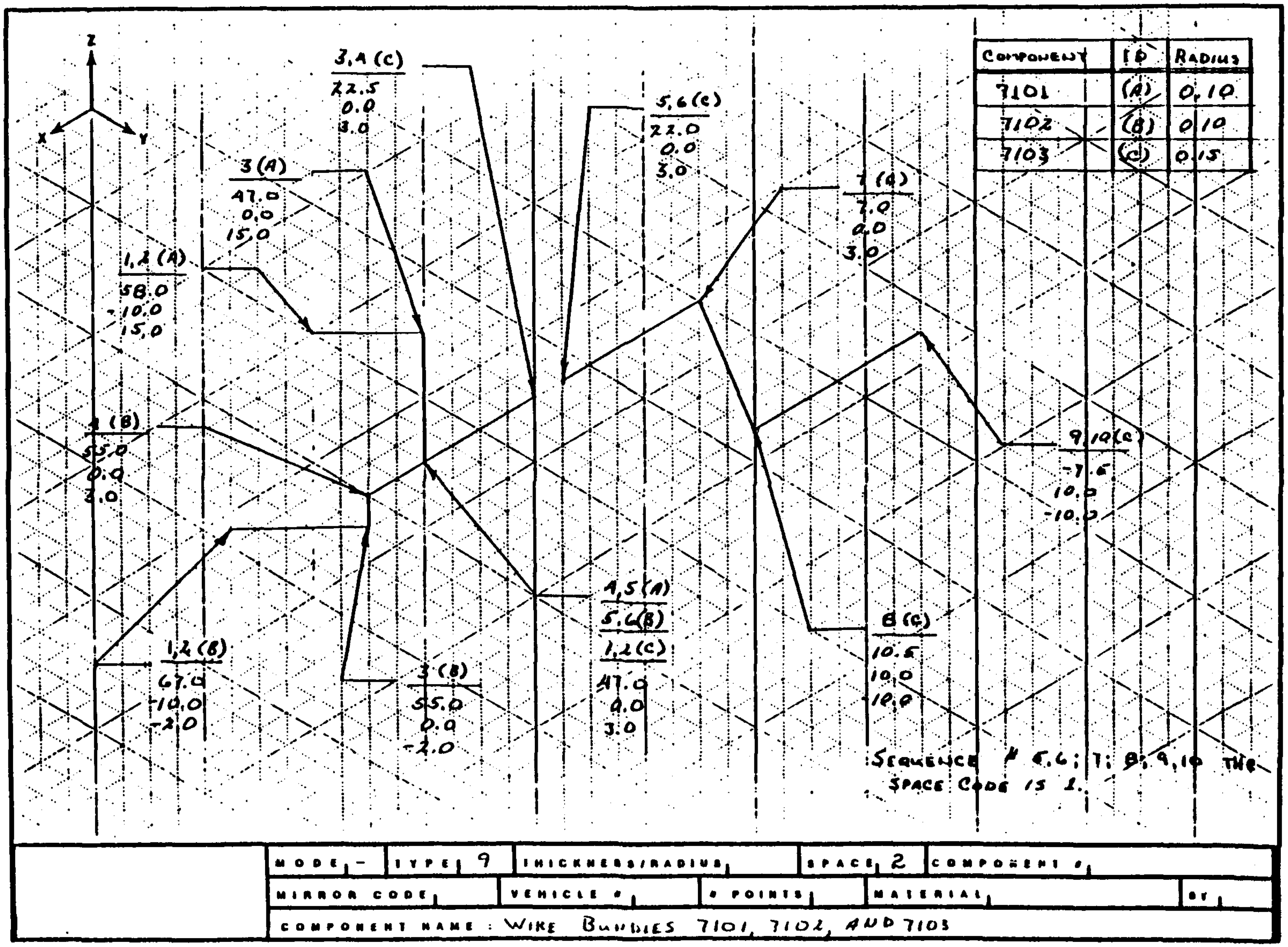


Figure 36. Wires Modeled in the Rod Mode

X COORD OR RADIUS	Y COORD OR RADIUS	Z COORD	±GNNS	COMP CODE	SEQUENCE NUMBER	ADDITIONAL SEQUENCE NUMBERS							MIRROR	VC
00000000 12345678	01111111 90123456	111222222 789012345	222233 678901	3333 2345	33334444444 67890123456	4445 7890	5555 1234	5555 5678	5666 9012	6666 3456	6667 7890	7777 1234	777 567	778 890
58.00	-10.00	15.00	-9102	7101	10	20								999
47.00	0.00	15.00	-9102	7101	30									999
47.00	0.00	3.00	-9102	7101	40	50								999
67.00	-10.00	-2.00	-9102	7102	10	20								999
55.00	0.00	-2.00	-9102	7102	30									999
55.00	0.00	3.00	-9102	7102	40									999
47.00	0.00	3.00	-9102	7102	50	60								999
47.00	0.00	3.00	-9102	7103	10	20								999
22.50	0.00	3.00	-9102	7103	30	40								999
22.00	0.00	3.00	-9102	7103	50	60								999
7.00	0.00	3.00	-9102	7103	70									999
10.50	10.00	-10.00	-9102	7103	80									999
-7.50	10.00	-10.00	-9102	7103	90	100								999

Figure 37. Coding for Three Components Described in Rod Mode

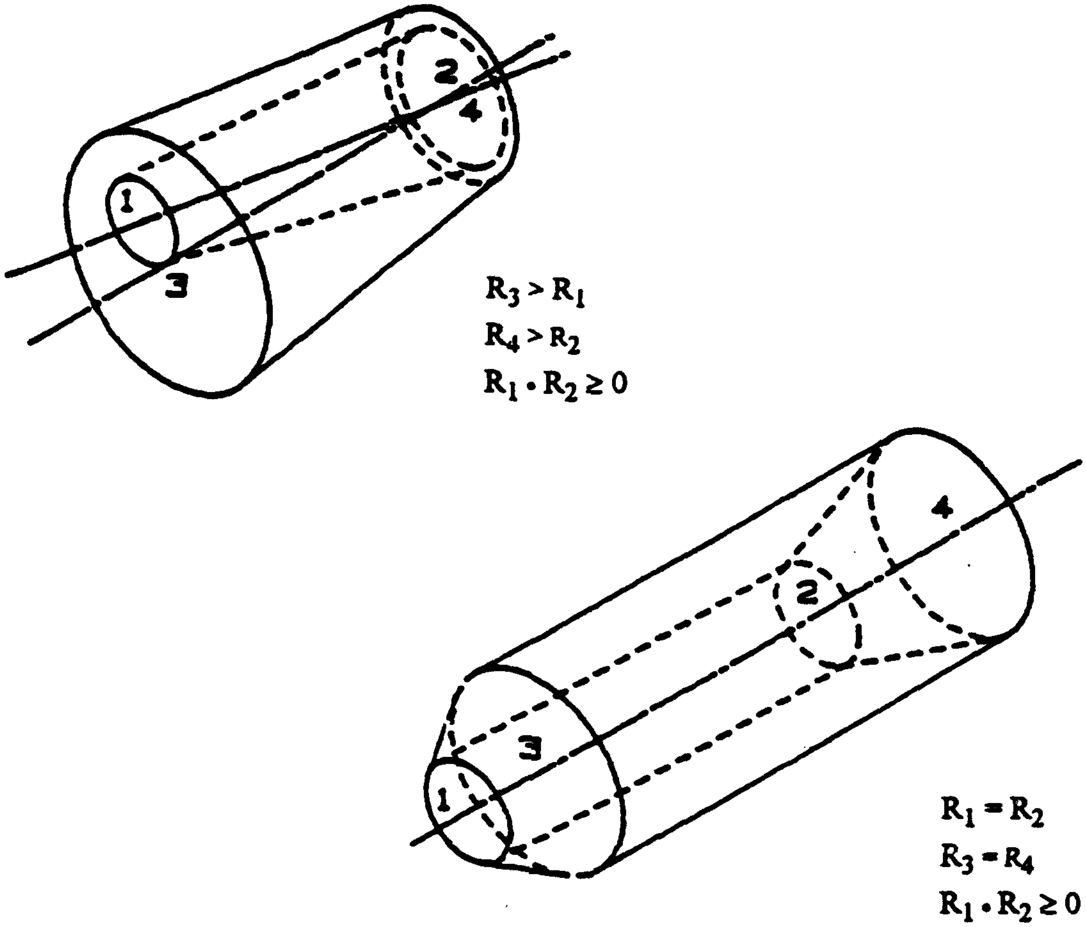
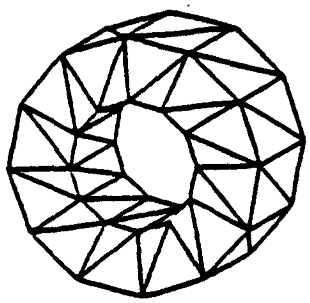


Figure 38. Illustration of Shapes Modeled Using Donut Primitive

TABLE 7. END CODES FOR DONUT PRIMITIVES FOR PLATE MODE

CODE	DESCRIPTION (Plate mode only)
0*	Both ends A and B open
1	End A open and end B closed
2	End B open and end A closed
3	Both ends A and B closed
4	End A disk only
5	End B disk only
6	Both end A and end B disks
* → In volume mode, both ends closed	



Volume Mode, Code 0
Plate Mode, Code 3

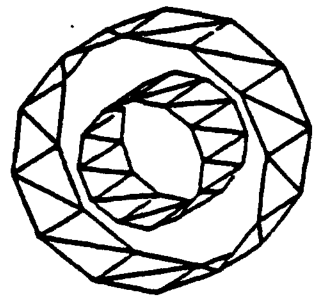


Plate Mode, Code 0

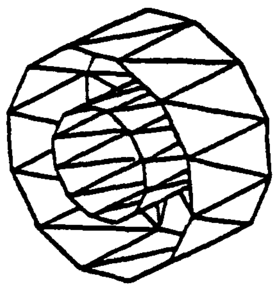


Plate Mode, Code 1 or 2

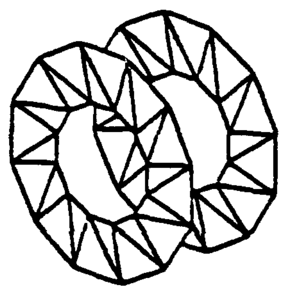


Plate Mode, Code 6

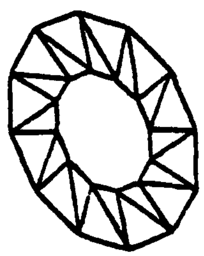


Plate Mode, Code 4 or 5

Figure 39. The Effects of End Point Coding on Donut Primitive

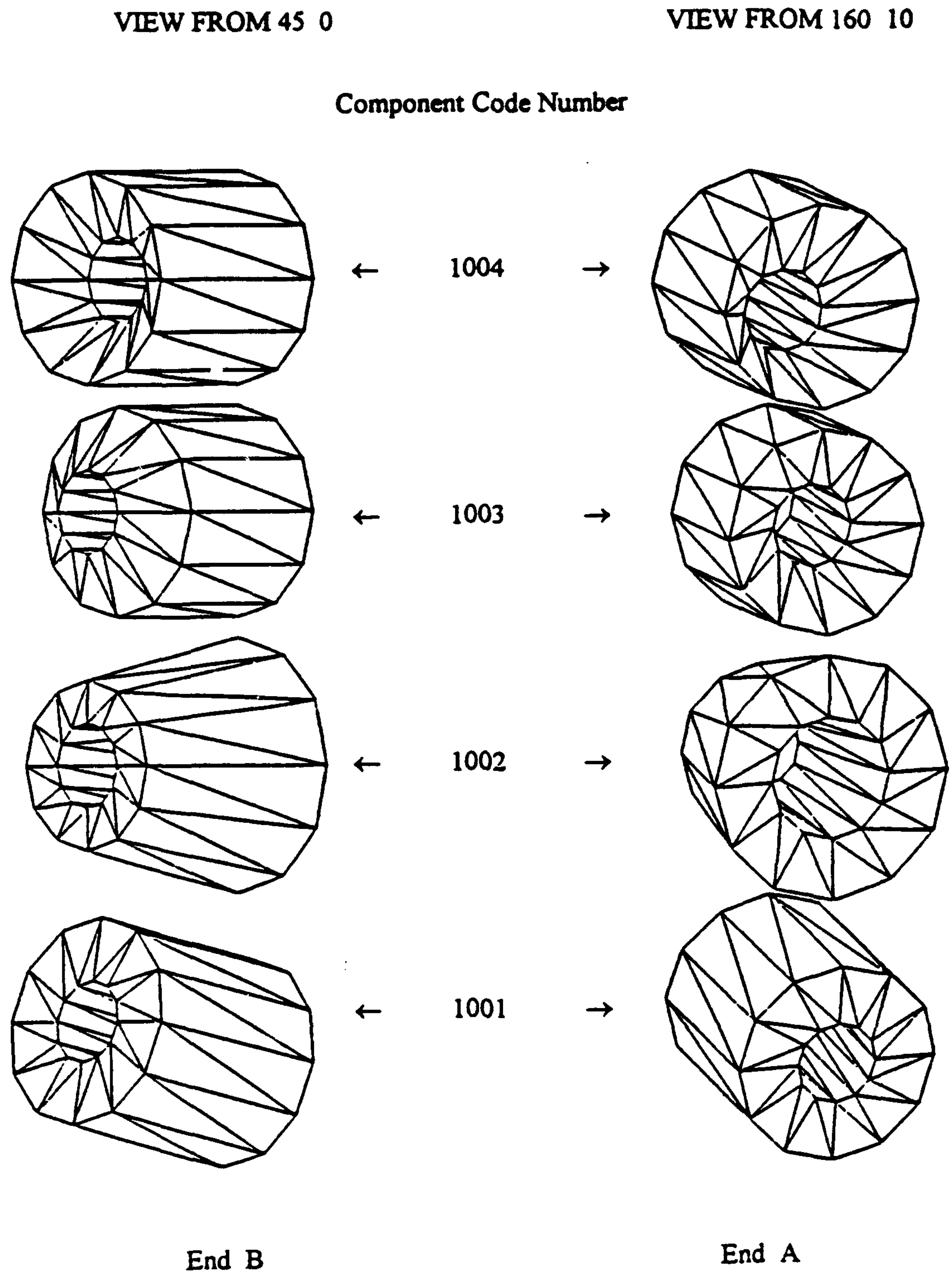


Figure 40. Two Views of Four Components Modeled With Donut Primitives in Volume Mode

X COORD OR RADIUS	Y COORD OR RADIUS	Z COORD	±GNNS	COMP CODE	SEQUENCE NUMBER	ADDITIONAL SEQUENCE NUMBERS							MIRROR	VC
00000000 12345678	01111111 90123456	111222222 789012345	222233 678901	3333 2345	33334444444 67890123456	4445 7890	5555 1234	5555 5678	5666 9012	6666 3456	6667 7890	7777 1234	777 567	778 890
0.00	0.00	4.00	3001	1001	10									999
10.00	0.00	6.00	3001	1001	20									999
5.00	5.00	0.00	3001	1001	30									999
0.00	0.00	4.00	3001	1001	40									999
10.00	0.00	6.00	3001	1001	50									999
2.00	2.00	0.00	3001	1001	60									999
0.00	0.00	18.00	3001	1002	10									999
10.00	0.00	18.00	3001	1002	20									999
6.00	4.00	0.00	3001	1002	30									999
0.00	0.00	18.00	3001	1002	40									999
10.00	0.00	18.00	3001	1002	50									999
3.00	2.00	0.00	3001	1002	60									999
0.00	0.00	30.00	3001	1003	10									999
8.00	0.00	30.00	3001	1003	20									999
5.00	5.00	0.00	3001	1003	30									999
2.00	0.00	30.00	3001	1003	40									999
10.00	0.00	30.00	3001	1003	50									999
2.00	2.00	0.00	3001	1003	60									999
0.00	0.00	41.00	3001	1004	10									999
10.00	0.00	41.00	3001	1004	20									999
5.00	5.00	0.00	3001	1004	30									999
0.00	2.00	41.00	3001	1004	40									999
10.00	2.00	41.00	3001	1004	50									999
2.00	2.00	0.00	3001	1004	60									999

Figure 41. Coding for Four Donut Primitives

3. MIRRORED COMPONENTS

Components may be "mirrored" across the X Z target plane using the CONVERT program. When this option is selected for a component, the program changes the sign of the Y coordinates of the selected components, retains the X and Z coordinates, and produces a mirrored component while retaining the original component as illustrated in Figure 42. This feature significantly reduces the modeling effort for most targets, particularly aircraft.

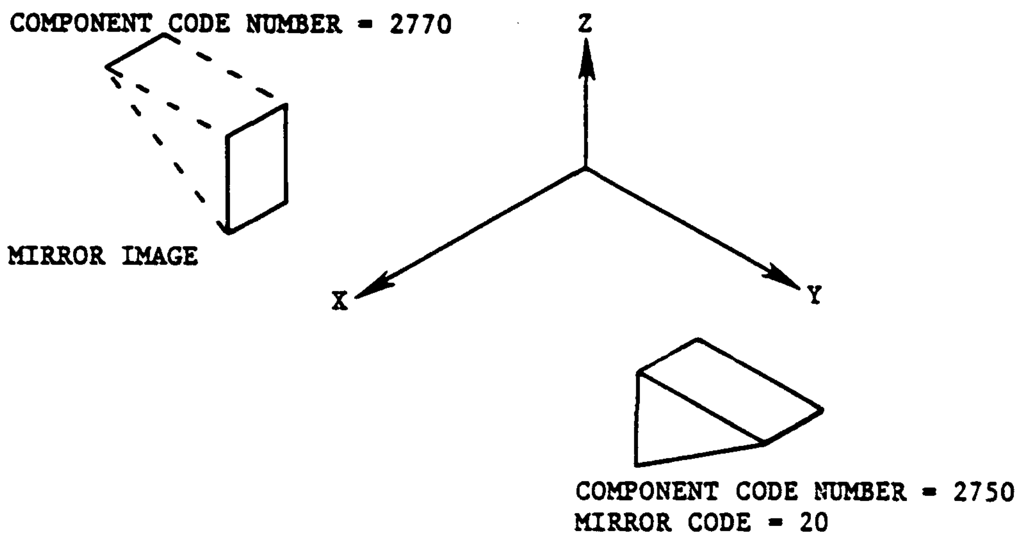


Figure 42. Component Mirror Image

To generate a mirrored component, the modeler need only enter a component code increment number in columns 75 through 77 of the data card for each set of sequence points listed for the component to be mirrored. For example, if the component code number for a mirrored component were 2750, entering 20 in columns 76 through 77 would generate a mirrored component with a component code number of 2770. Care must be taken not to generate a component code number already in use. Figures 43 and 44 illustrate a mirrored component and corresponding data card entries.

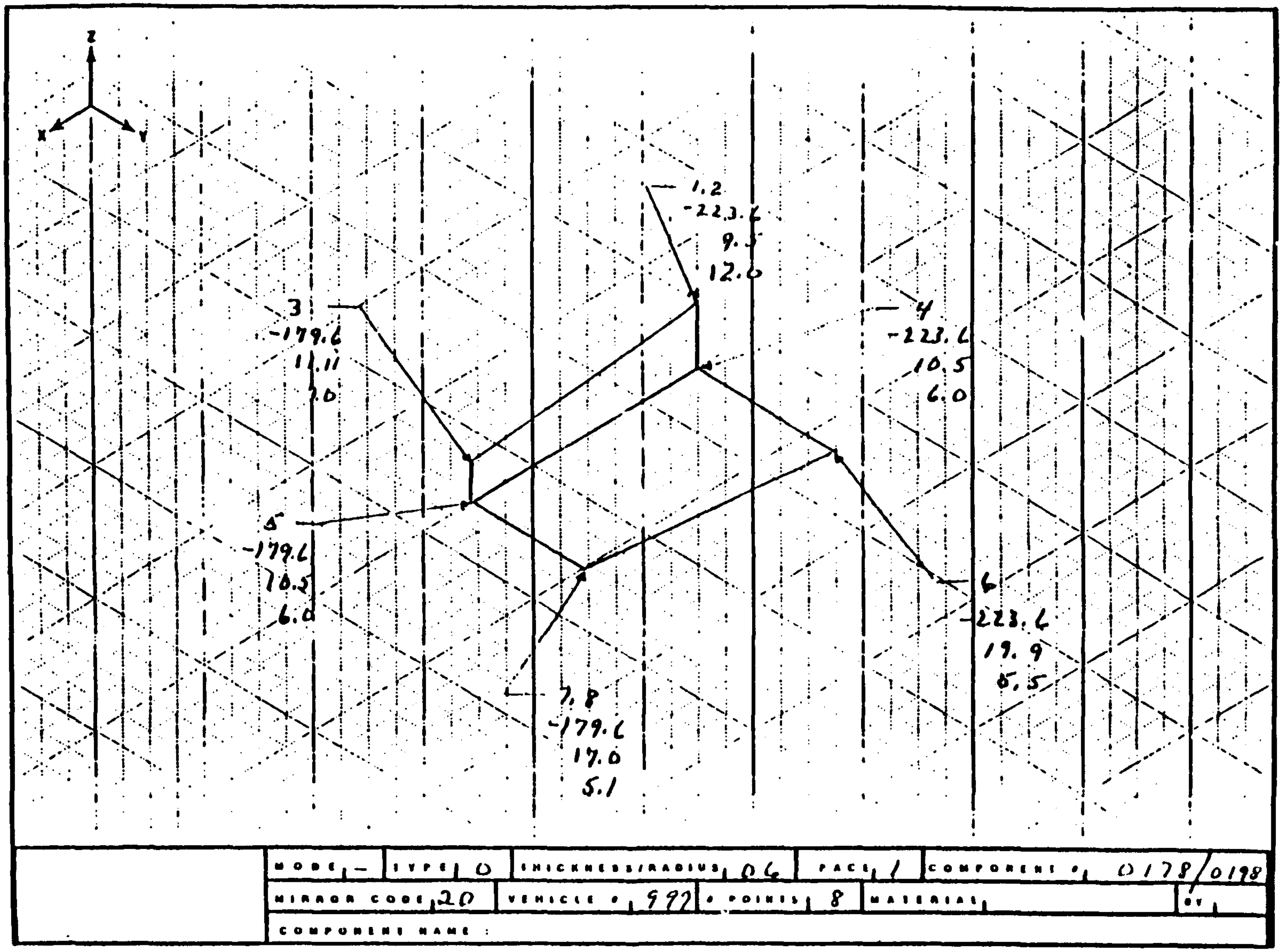


Figure 43. Sketch of Mirrored Component With Different Code Number

X COORD OR RADIUS	Y COORD OR RADIUS	Z COORD	±GNHS	COMP CODE	SEQUENCE NUMBER	ADDITIONAL SEQUENCE NUMBERS							MIRROR	VC
00000000	01111111	111222222	222233	3333	33334444444	4445	5555	5555	5666	6666	6667	7777	777	778
12345678	90123456	789012345	678901	2345	67890123456	7890	1234	5678	9012	3456	7890	1234	567	890
-223.60	9.50	12.00	-0061	0178	10	20							20	999
-179.6	11.11	7.00	-0061	0178	30								20	999
-223.60	10.50	6.00	-0061	0178	40								20	999
-179.60	10.50	6.00	-0061	0178	50								20	999
-223.60	19.90	5.00	-0061	0178	60								20	999
-179.60	17.00	5.10	-0061	0178	70	80							20	999

Figure 44. Coding to Create a Mirrored Component With a Different Code Number

The mirrored component option can also be used to describe the remaining part of a component only half described. This is applicable to components that are symmetrical to, and lie along, the longitudinal axis of the target such as the canopy of an aircraft as illustrated in Figure 45. The code entered in columns 75 through 77 for this type of mirroring is -1 which causes the mirrored half of the component to retain the component code number of the half being mirrored. Figures 46 and 47 illustrate a section of fuselage and related coding for mirroring in this fashion.

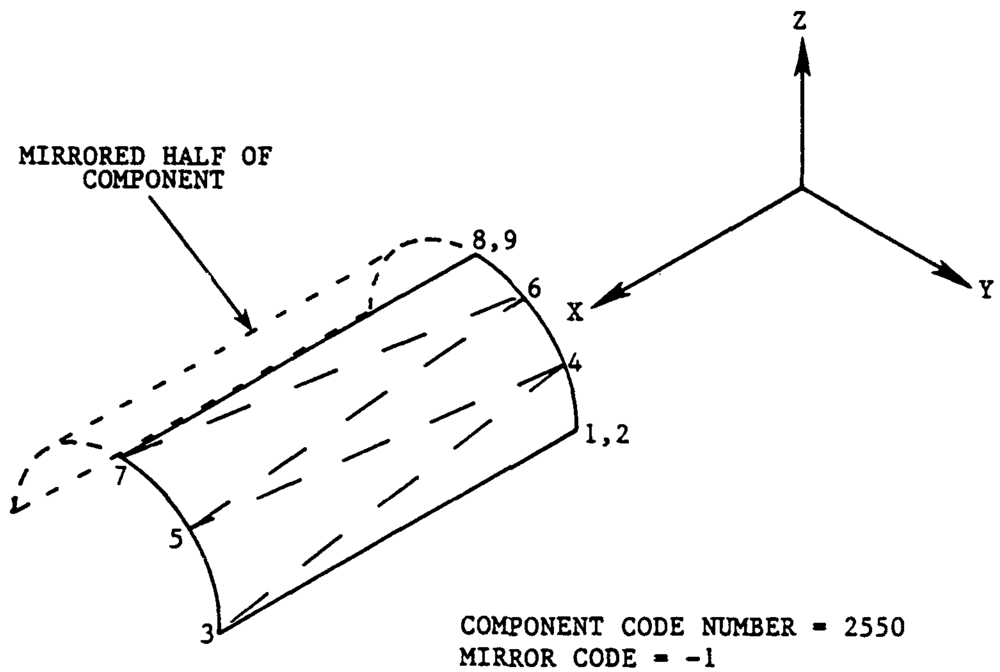


Figure 45. Canopy-Type Component Illustrating the Mirror Modeling Techn que

4. VOLUME SUBTRACTION

Volume subtraction permits the removal of a portion of a component described in volume mode, by describing another volume mode component having the same component code number. For example, consider the component illustrated in Figure 48. The component could be modeled by describing a large number of triangles to cover both the outside surfaces of the component and the inner surfaces forming the opening. Volume subtraction offers a much simpler method. If both the enclosing volume and the opening are described as boxes (four records each) in volume mode, and if both primitives are assigned the same component code number, then the smaller box would generate an opening in the larger box. Figure 49 illustrates the descriptive coding for this component.

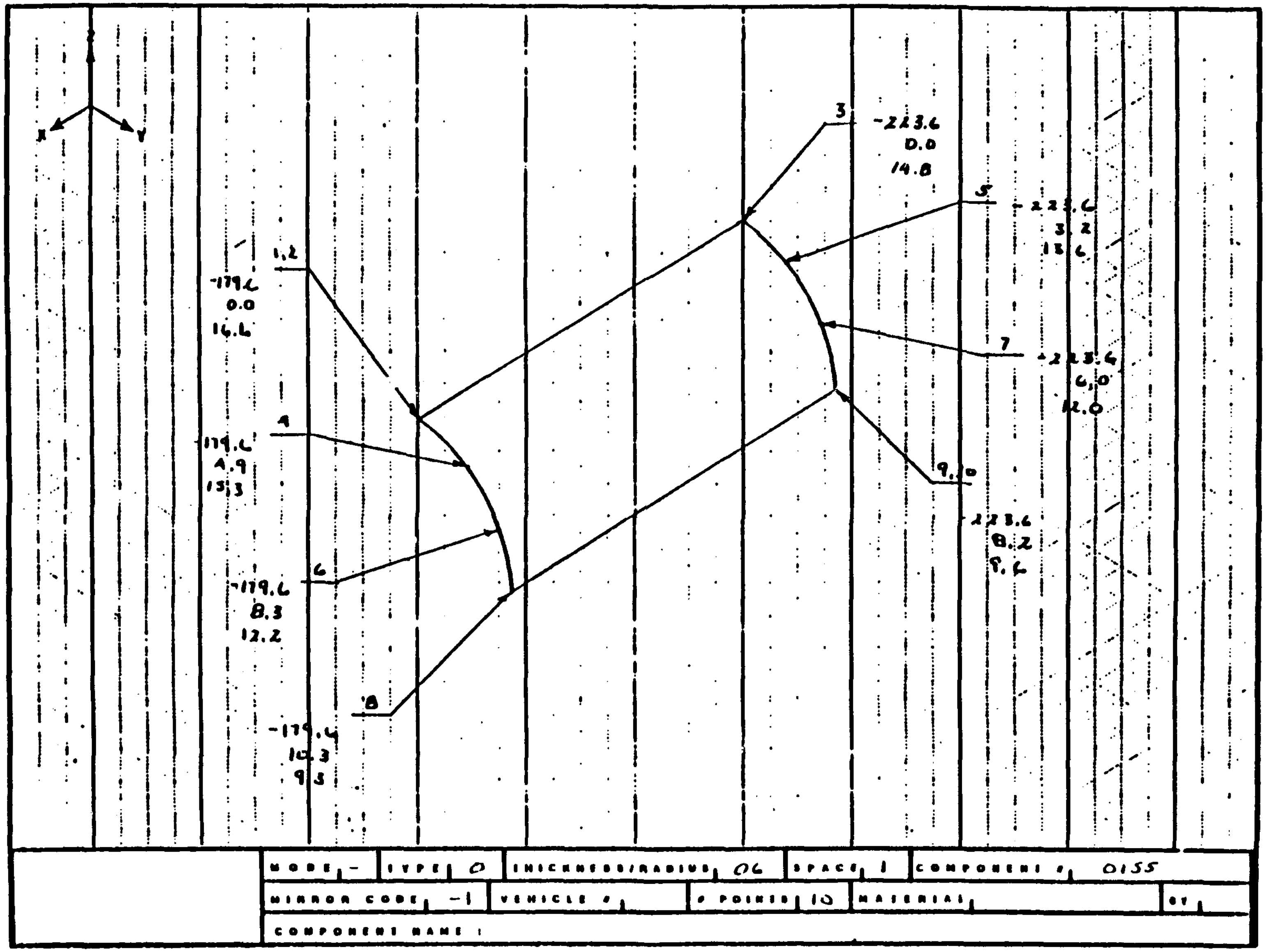


Figure 46. Mirrored Component With Same Code Number

X COORD OR RADIUS	Y COORD OR RADIUS	Z COORD	±GNNS	COMP CODE	SEQUENCE NUMBER	ADDITIONAL SEQUENCE NUMBERS							MIRROR	VC
00000000 12345678	01111111 90123456	111222222 789012345	222233 678901	3333 2345	33334444444 67890123456	4445 7890	5555 1234	5555 5678	5666 9012	6666 3456	6667 7890	7777 1234	777 567	778 890
-179.60	0.00	16.60	-0061	0155	10	20							-1	999
-223.60	0.00	14.80	-0061	0155	30								-1	999
-179.60	4.90	15.30	-0061	0155	40								-1	999
-223.60	3.20	13.60	-0061	0155	50								-1	999
-179.60	8.30	12.20	-0061	0155	60								-1	999
-223.60	6.00	12.00	-0061	0155	70								-1	999
-179.60	10.30	9.30	-0061	0155	80								-1	999
-223.60	8.20	9.60	-0061	0155	90	100							-1	999

Figure 47. Coding to Create a Mirrored Component With the Same Code Number

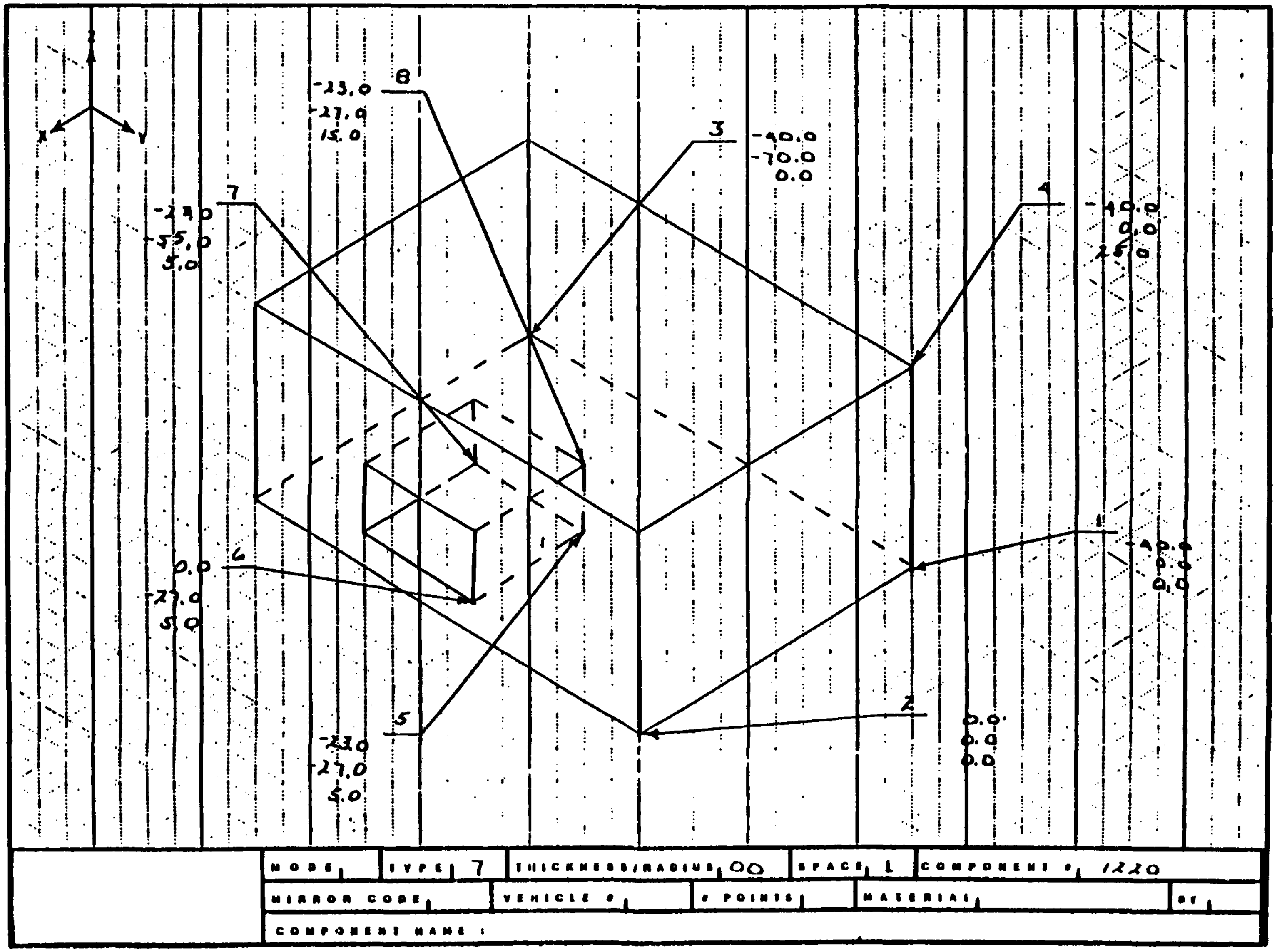


Figure 48. Sketch of Component With an Opening Created by Volume Subtraction

X COORD OR RADIUS	Y COORD OR RADIUS	Z COORD	±GMNS	COMP CODE	SEQUENCE NUMBER	ADDITIONAL SEQUENCE NUMBERS							MIRROR	VC
00000000	01111111	111222222	222233	3333	33334444444	4445	5555	5555	5666	6666	6667	7777	777	778
12345678	90123456	789012345	678901	2345	67890123456	7890	1234	5678	9012	3456	7890	1234	567	890
-40.00	0.00	0.00	7001	1220	10									999
0.00	0.00	0.00	7001	1220	20									999
-40.00	-70.00	0.00	7001	1220	30									999
-40.00	0.00	25.00	7001	1220	40									999
-23.00	-27.00	5.00	7001	1220	50									999
0.00	-27.00	5.00	7001	1220	60									999
-23.00	-55.00	5.00	7001	1220	70									999
-23.00	-27.00	15.00	7001	1220	80									999

Figure 49. Coding of a Component With an Opening Created by Volume Subtraction

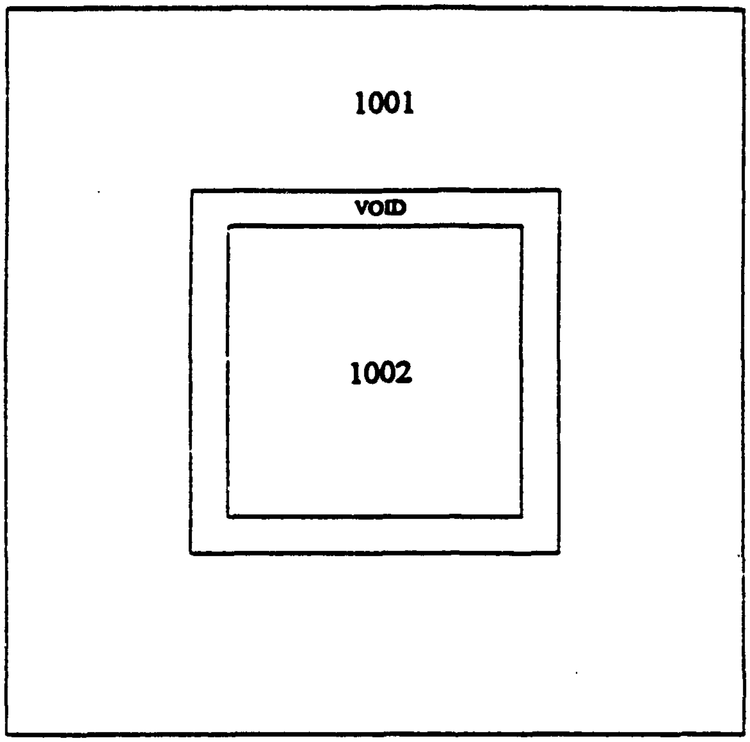
Besides providing an opening, volume subtraction will allow removal of an interior portion of a component modeled in volume mode and the insertion of a component of a different name into the vacated space. For example, Figure 50(a) shows a PIXPL plot of a box modeled in volume mode. The interior of the box was removed by defining a smaller box having the same component code number as the outer box. Then a third box with a different component code number (1002) was defined and located within the vacated space. The PIXPL plot was made by defining a clipping plane which cut through all three boxes allowing one to view the interior. Volume subtraction is useful if one considers a fuel pump located inside a fuel cell. One method of modeling these components would be to describe the fuel as one volume with a void within the fuel created by volume subtraction. Then model the fuel pump as a separate component within the void.

Figure 50(b) shows a PIXPL plot of the interior of a box modeled in volume mode where volume subtraction was accomplished using a sphere primitive. The coding for volume subtraction of the components listed in Figure 50 is shown in Figure 51.

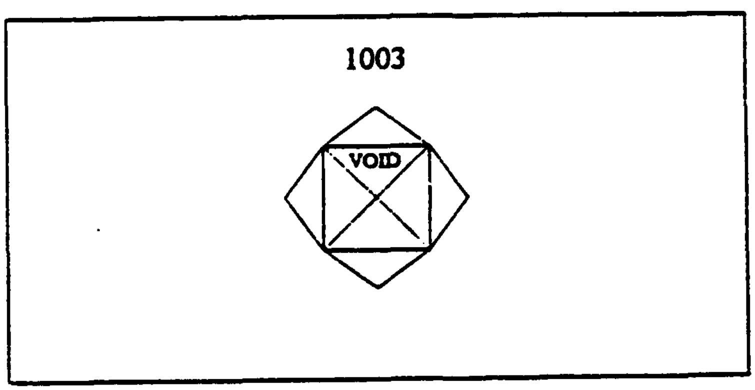
5. FUEL TANK LEVELS

Often the vulnerability analyst is required to make an assessment as to whether or not a particular aircraft would be able to complete its mission, given that the aircraft was attacked somewhere between takeoff and the target area. To perform such an assessment, the analyst is required to make certain assumptions concerning the fuel levels remaining in each of the aircraft fuel tanks at the time of the attack. The analyst needs this information because the damage that results from perforating a full tank differs significantly from the damage that occurs when perforating the void space (ullage) of a partially filled tank. For this reason, the fuel in aircraft tanks is usually modeled in levels that indicate the percentage of the total tank volume. For example, the modeler might model the bottom level up to 25 percent of capacity, the next level up to 50 percent, the third level up to 75 percent, and the last level at the top of the tank as shown in Figure 52. The fuel levels would be defined in the component code list in a manner such as:

Component Code	Description	Material	Density
4055	0 - 25% Fuel in Left Wing Tank	JP4	100
4056	25 - 50% Fuel in Left Wing Tank	JP4	100
4057	50 - 75% Fuel in Left Wing Tank	JP4	100
4058	75 - 100% Fuel in Left Wing Tank	JP4	100



(a)



(b)

Figure 50. (a) A Box Inside a Box and (b) A Box With a Sphere-Shaped Void

X COORD OR RADIUS	Y COORD OR RADIUS	Z COORD	±GNNS	COMP CODE	SEQUENCE NUMBER	ADDITIONAL SEQUENCE NUMBERS							MIRROR	VC
00000000 12345678	01111111 90123456	111222222 789012345	222233 678901	3333 2345	33334444444 67890123456	4445 7890	5555 1234	5555 5678	5666 9012	6666 3456	6667 7890	7777 1234	777 567	778 890
0.00	0.00	0.00	7001	1001	10									
20.00	0.00	0.00	7001	1001	20									
0.00	20.00	0.00	7001	1001	30									
0.00	0.00	20.00	7001	1001	40									
5.00	5.00	5.00	7001	1001	50									
15.00	5.00	5.00	7001	1001	60									
5.00	15.00	5.00	7001	1001	70									
5.00	5.00	15.00	7001	1001	80									
6.00	6.00	6.00	7001	1002	10									
14.00	6.00	6.00	7001	1002	20									
6.00	14.00	6.00	7001	1002	30									
6.00	6.00	14.00	7001	1002	40									
21.00	0.00	0.00	7001	1003	10									
31.00	0.00	0.00	7001	1003	20									
21.00	20.00	0.00	7001	1003	30									
21.00	0.00	20.00	7001	1003	40									
26.00	10.00	10.00	6001	1003	50									
2.50	2.50	0.00	6001	1003	60									

Figure 51. Coding to Create a Box-Shaped Void and a Sphere-Shaped Void Using Volume Subtraction

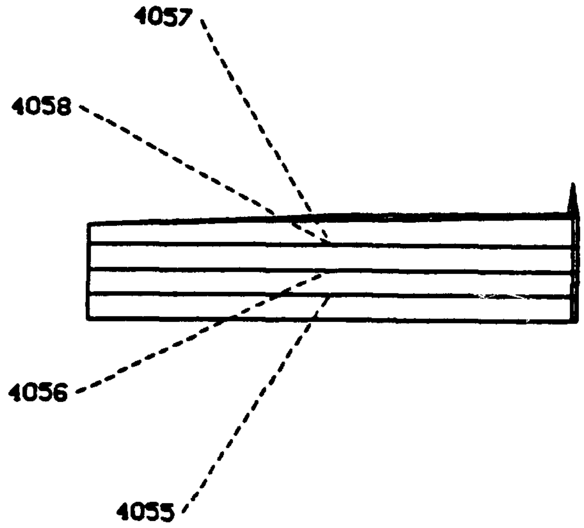


Figure 52. Wing Tank Modeled With Four Levels of Fuel

If the analyst decides that the fuel tank should be half full, then a density of zero would be assigned to the ullage of components 4057 and 4058 and the proper damage functions defined for all four levels of fuel.

6. PLATE MODE ASYMMETRY AND SIDE EFFECTS

The manner in which LOS thickness is determined for plate mode components warrants separate consideration. When a shotline encounters - volume mode component, as illustrated in Figure 53(a), the LOS thickness is calculated based on the difference between the entry and exit intersections of the component, as measured along the shotline. Intersections are considered in pairs. However, plate mode descriptions are approximations and as such, may cause some difficulty if not completely understood. In Figure 53(b), shotlines C1 and C2 represent two shotlines traveling in opposite directions and hitting a plate mode component. The data defines the first surface encountered by the shotline in the direction of travel. Since plate mode components are defined by a surface and a thickness, the implied location of the second surface depends on the direction of the shotline. Figure 54 illustrates the potential problem this might create if a thick plate component were modeled in plate mode with other components sharing the surface of the thick plate. To avoid this situation, the modeler should attempt to provide some separation between a component modeled in plate mode and other components.

Another potential problem arises when the shotline is not perpendicular to the surface as illustrated in Figure 53(c). C1 is not recorded as an intersection because the ray does not

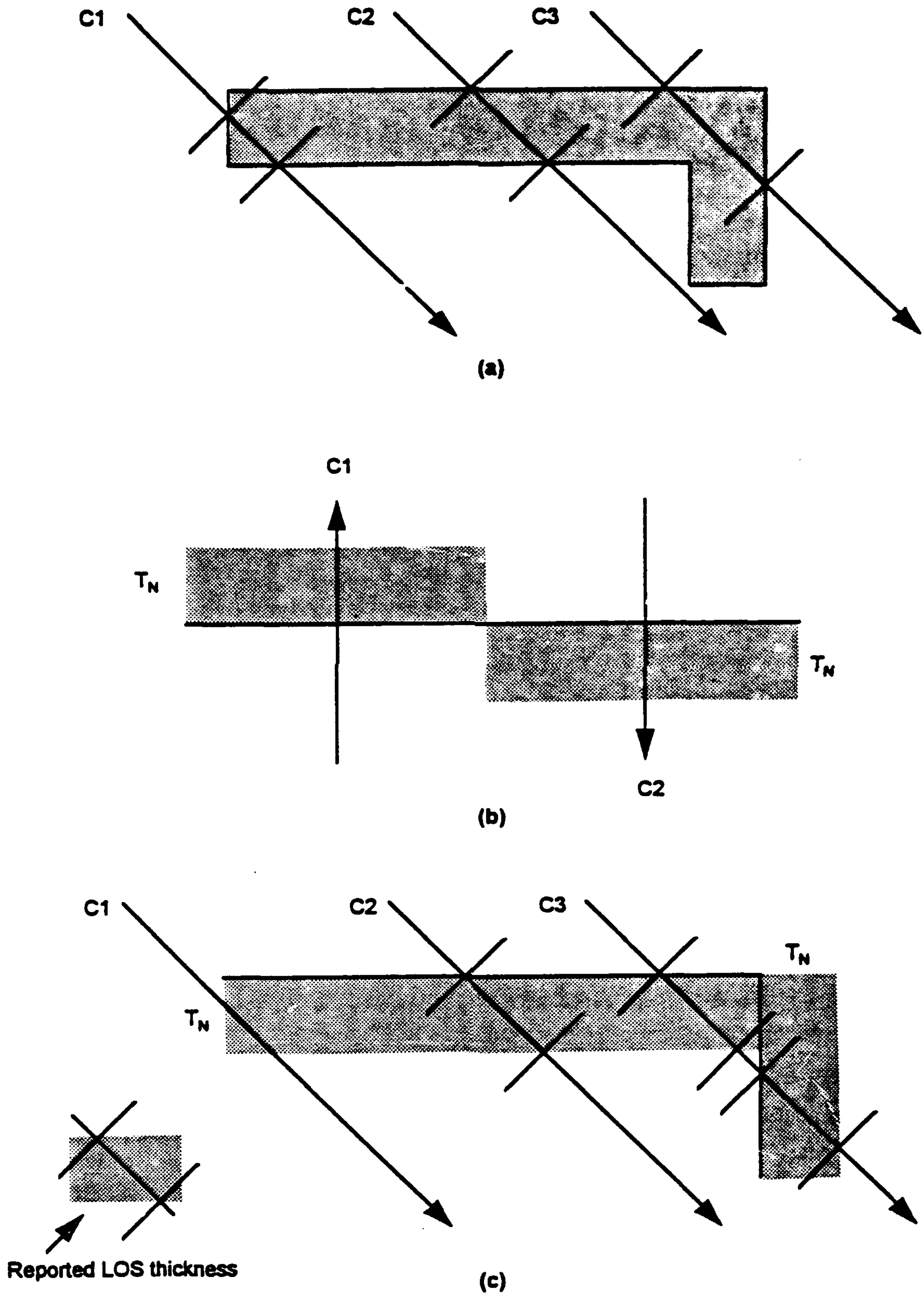


Figure 53. The Asymmetrical and Edge Effects of Plate Mode Components

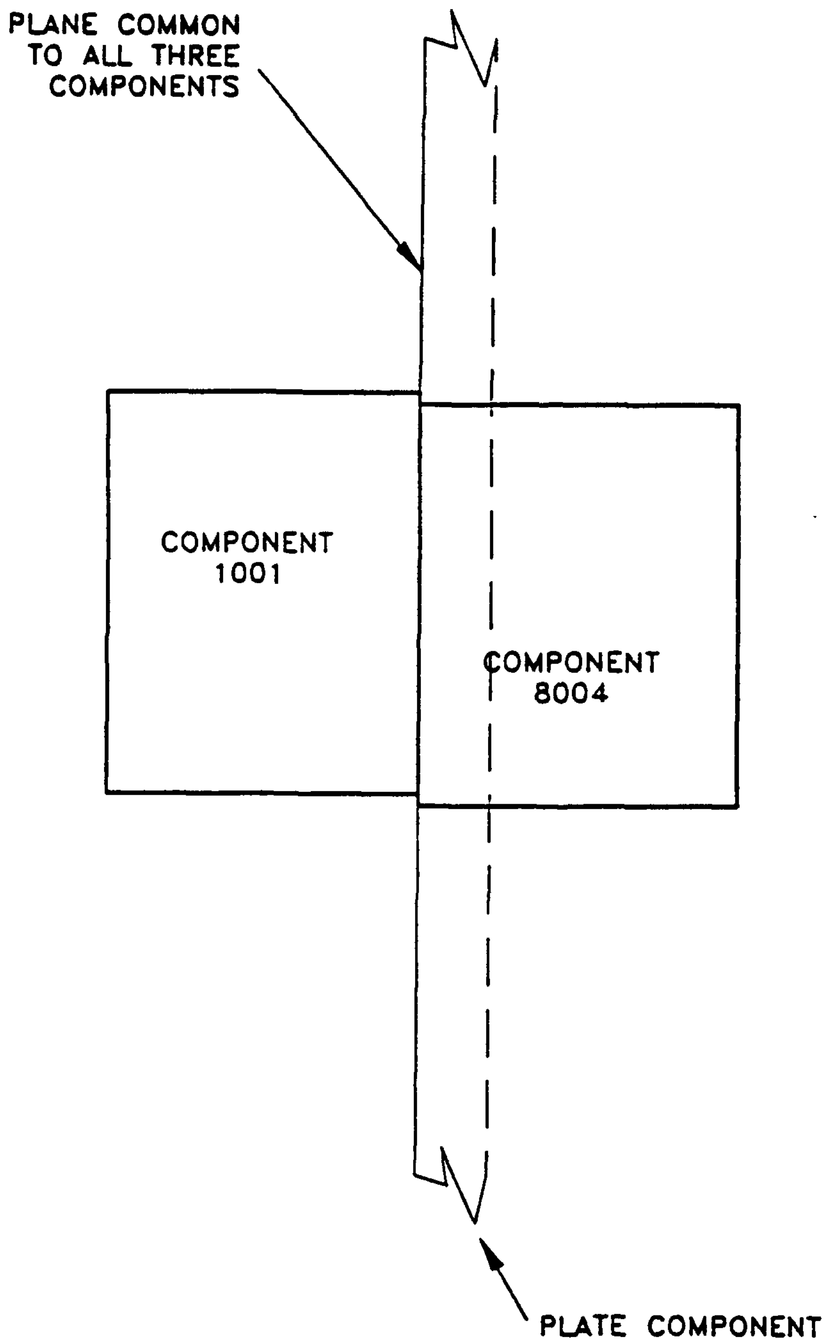


Figure 54. The Asymmetrical Effects of a Plate Mode on Other Components

intersect the first surface. C2 is correct except for asymmetry. C3 overstates the proper LOS distance. The magnitude of the errors increases with increasing shotline obliquity and increasing plate thickness. As these two effects can combine, it is recommended that the thickness of components, modeled in plate mode, be limited to a maximum of 0.50-inch.

7. INTERFERENCE

Interference occurs when part or all of one component is modeled such that it occupies the space of a component modeled in volume mode. There are three types of interference possible. They are termed as ambiguous interference, unallowable interference, and intentional interference.

a. Ambiguous Interference

Ambiguous interference occurs when a shotline detects a portion of a volume mode component that is not totally within the volume space of another volume mode component. When a shotline enters a volume mode component, it is expected that an entrance surface and an exit surface will be encountered before encountering an intervening surface of another component modeled in volume mode. If an intervening surface is encountered, the region common to both volumes is ambiguous and FASTGEN prints an error digit, "4".

Ambiguous interference can occur when one models the adjoining surfaces of two components modeled in volume mode such as the surfaces of two levels of fuel or two sections of a tank turret. To avoid the possibility of generating an ambiguous interference condition, one should assure adjacent triangulated surfaces of the adjoining surfaces "match" to avoid subsequent interference.

b. Unallowable Interference

Unallowable interference occurs whenever a component is detected inside a volume mode component. The interfering component may be modeled in volume, plate, or rod mode. Such interference frequently occurs because one component was inadvertently placed too close to another. Figure 55(a) shows a rod mode component modeled such that one end (P) terminates on the surface, and at an angle other than normal to the surface, of a volume mode component. The figure illustrates how unallowable interference may occur due to the radius (R) of the rod. To avoid such interference, a gap approximately equal to the rod radius should be modeled between the surface of the volume mode component and

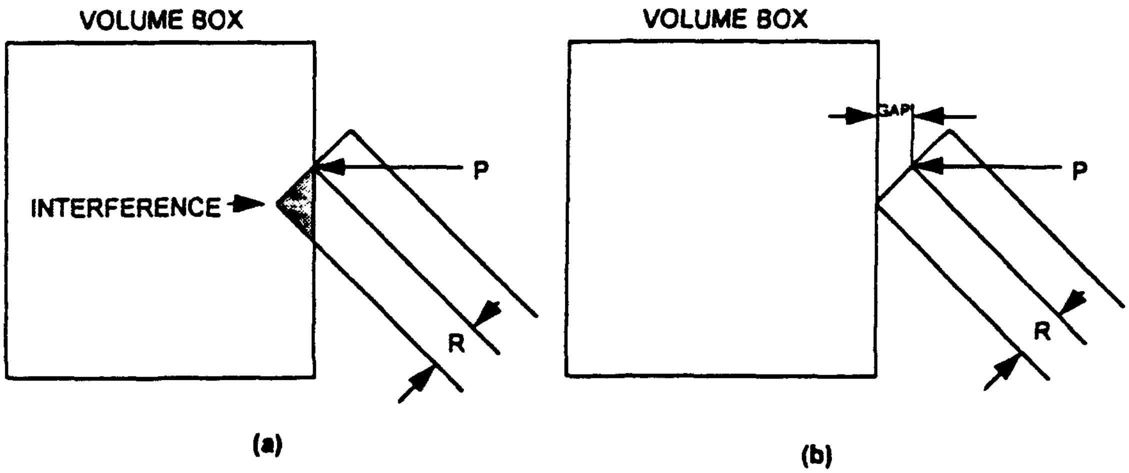


Figure 55. Unallowable Interference Between Rod and a Volume Mode Component

the rod end as illustrated in Figure 55(b). To avoid such interference, a gap at least equal to the radius of the rod should be modeled between the surface of the volume mode component and the rod end.

c. Intentional Interference

There are several occasions when it is desirable to model components, which otherwise would cause unallowable interference, as intentionally interfering components. The component that has other components or component parts described within its boundaries must be described in volume mode. The interfering components may be described in volume, plate, or rod mode. The FASTGEN user must identify the interfered component and the interfering components on the input specification RECORD TYPE 2. When FASTGEN processes shotlines through these components, it can adjust the shotline descriptions so that they properly represent the components and air spaces encountered by the shotline, and no error messages will be printed. For example, fuel in wing tanks is frequently described in volume mode with spars and ribs modeled within the fuel. Similarly, a fuel line routed through the fuel to a submerged fuel pump is frequently defined as interfering with the fuel.

Intentional interference must be used with care. Figure 56 illustrates some of the effects caused by declaring components as intentionally interfering components. The figure shows a large box (component 1001), 7-inches wide in the plan view, and modeled in volume mode. Modeled entirely within component 1001 are two smaller boxes; component 1002 modeled in volume mode and component 1003 modeled in plate mode. Components 1004 and 1005 are similarly modeled as boxes in volume mode and plate mode, respectively, but only

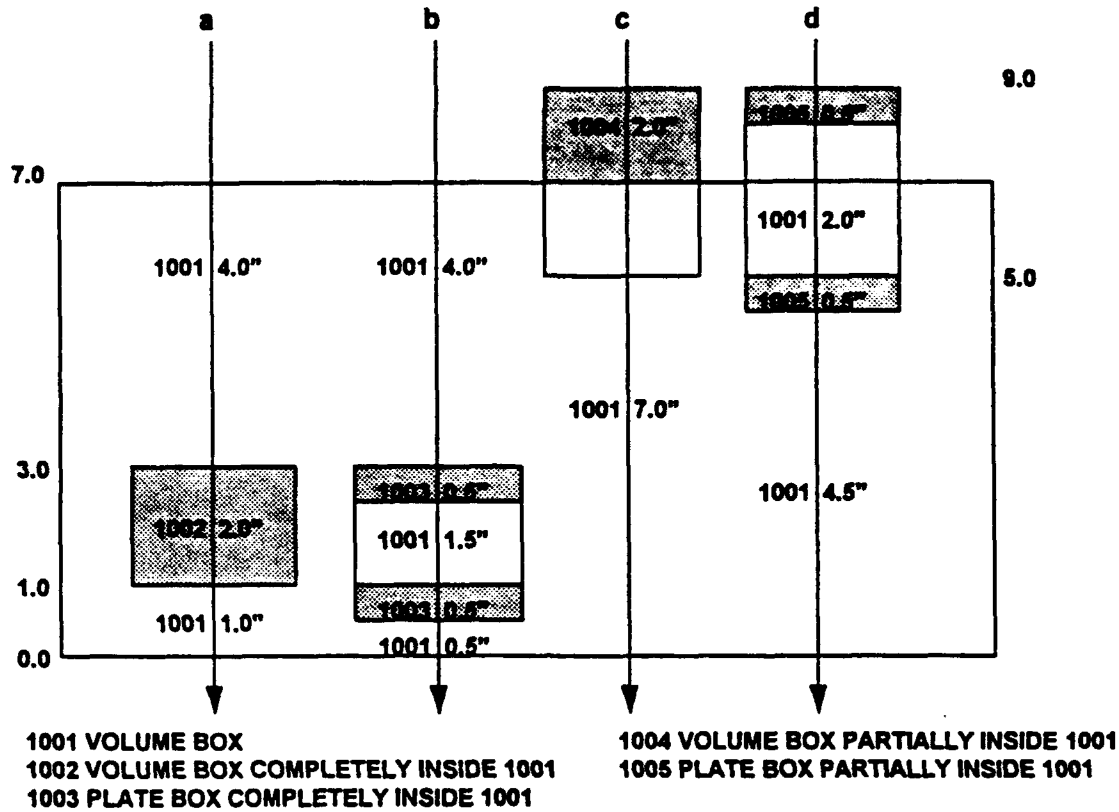


Figure 56. FASTGEN Interpretation of Some Intentionally Interfering Conditions

partially positioned inside the boundaries of component 1001. Components 1002, 1003, 1004, and 1005 were declared as intentionally interfering with component 1001. Shotlines a, b, c, and d intersected the components as shown by the arrowed lines in the figure. For these set of conditions, FASTGEN indicated the following:

Shotline	FASTGEN Interpretation
a	Correctly indicates 4-inches of component 1001, 2-inches of component 1002, and 1-inch of component 1001.
b	Correctly indicates 4-inches of component 1001, 0.5-inch for the two sides of component 1003, and 0.5-inch of component 1001. However, the interior of component 1003 was interpreted as 1.5 inches of component 1001. No error messages were issued.
c	Incorrectly interprets the data. FASTGEN reports 2 inches of component 1004 and then the total 7 inches of component 1001. Error messages indicating both unallowed and ambiguous interference were issued.
d	Correctly indicates 0.5-inch for the two sides of component 1005 and 4.5-inches for component 1001. However, the interior of the box positioned within component 1001 was interpreted as component 1001. No error messages were issued.

By examining the FASTGEN interpretations associated with each of the four shotlines, one can conclude the following:

- Shotline a shows that the placement of a volume component entirely within another volume component and declaring the former as interfering with the latter will be correctly interpreted by FASTGEN. For example, component 1001 could represent fuel and component 1002 could represent a fuel pump.
- Hollow components modeled in plate mode and located entirely within a volume mode component, as encountered by shotline b, are interpreted incorrectly by FASTGEN even though they are declared as intentionally interfering components. Had component 1003 been a solid component modeled in plate mode and declared as intentionally interfering with component 1001, FASTGEN would have interpreted its encounter on the shotline correctly.
- A shotline encounter with a volume mode component partially occupying the space of another volume mode component will always generate unallowed and/or ambiguous interference even though declared as intentionally interfering with one another as illustrated by shotline c.
- As with shotline b, the interior of component 1005 encountered by shotline d was incorrectly interpreted by FASTGEN to be part of component 1001.

8. SAMPLE CASES

Figures 57 through 62 show plots of selected groups of components generated from an aircraft TGM. Figures 63 through 68 contain reproductions of a few component sketches and are presented to illustrate the use of the various modeling constructs and their application in TGM development.

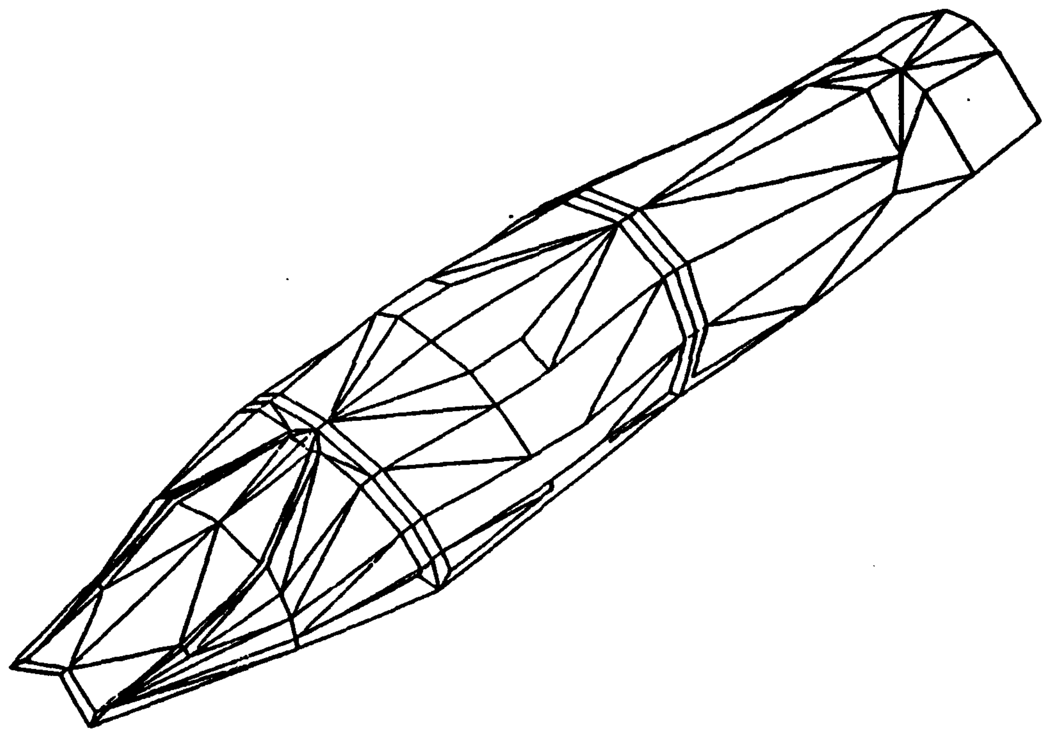


Figure 57. PIXPL Plot of an Aircraft Canopy

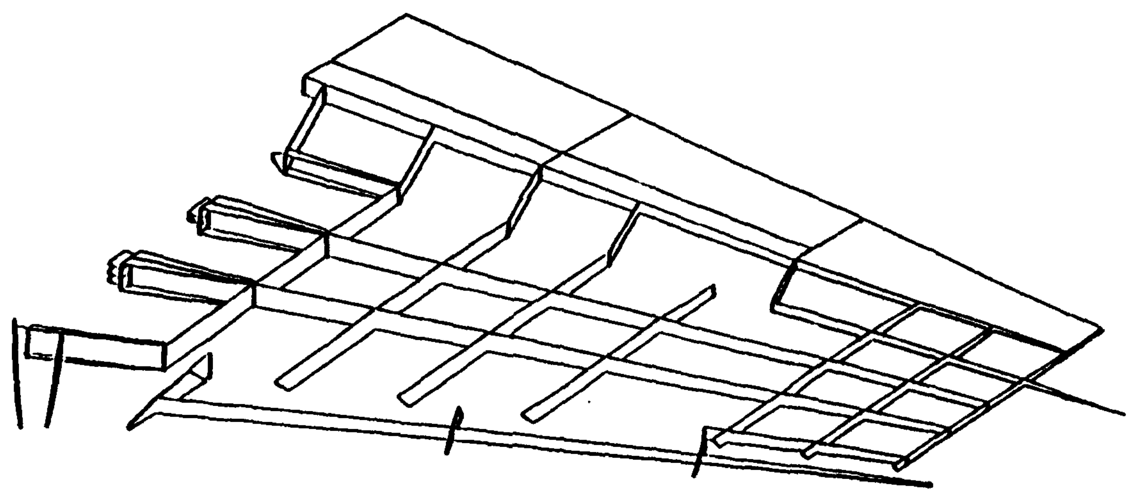


Figure 58. PIXPL Plot of Wing Spar and Rib Assembly

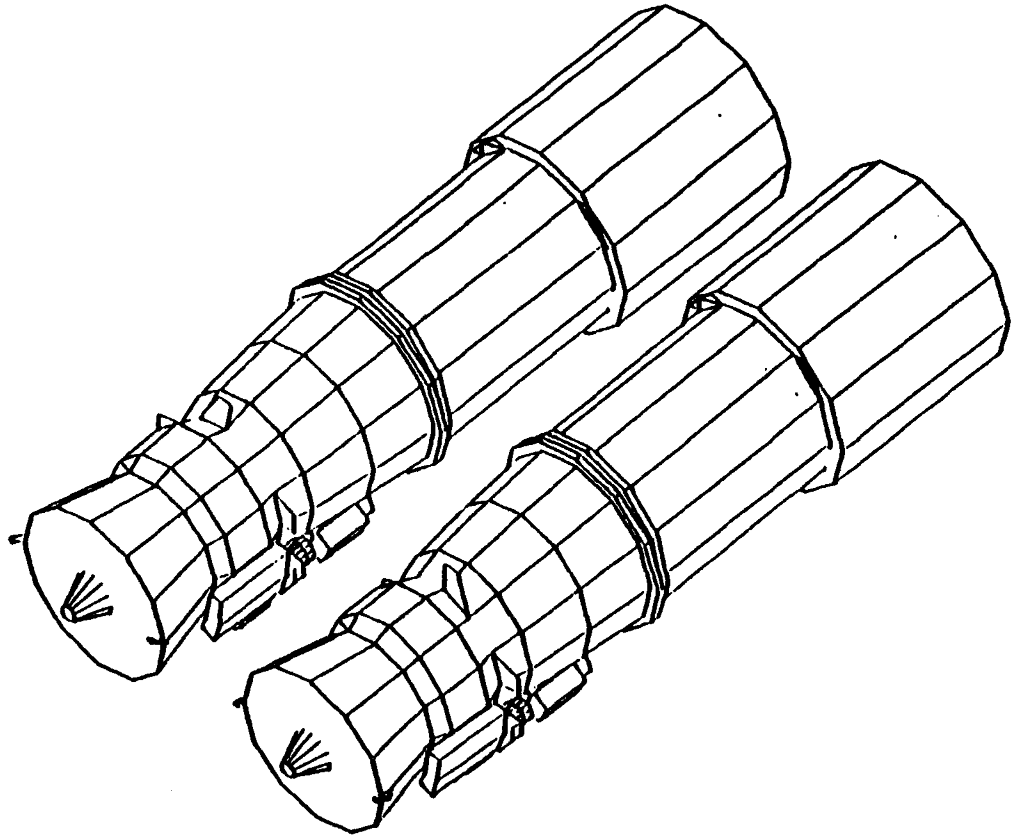


Figure 59. PIXPL Plot of Jet Engines Where One Engine Was Modeled as Mirror Image of the Other

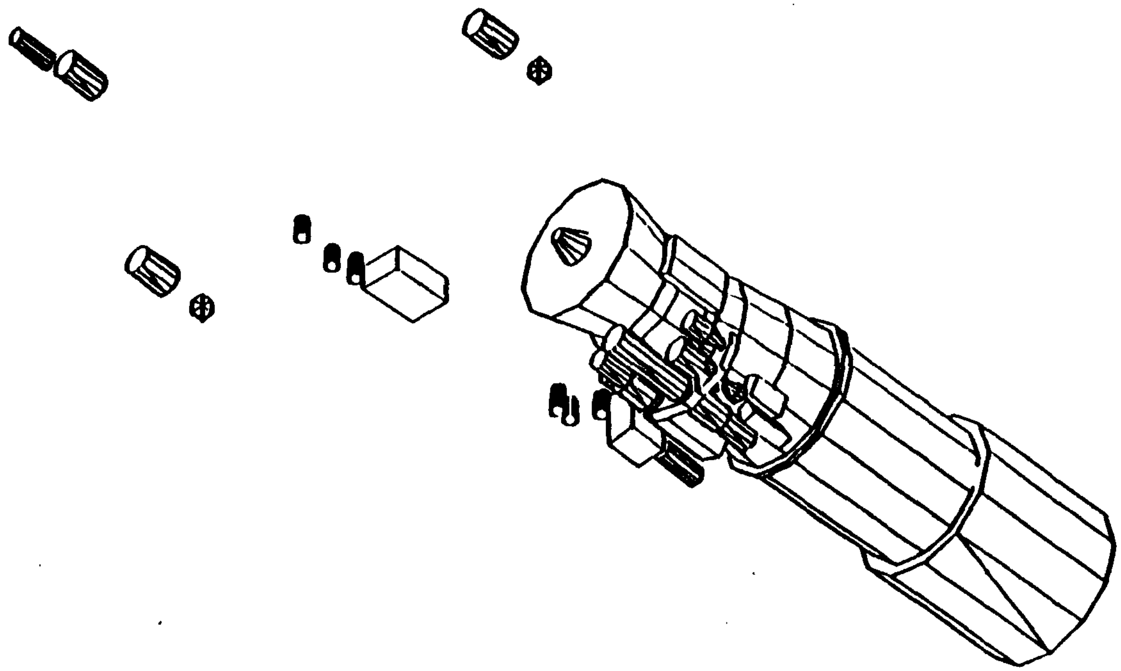


Figure 60. PIXPL Plot of Under Side of Jet Engine Showing Fuel Pumps, Generators, and Hydraulic Pumps

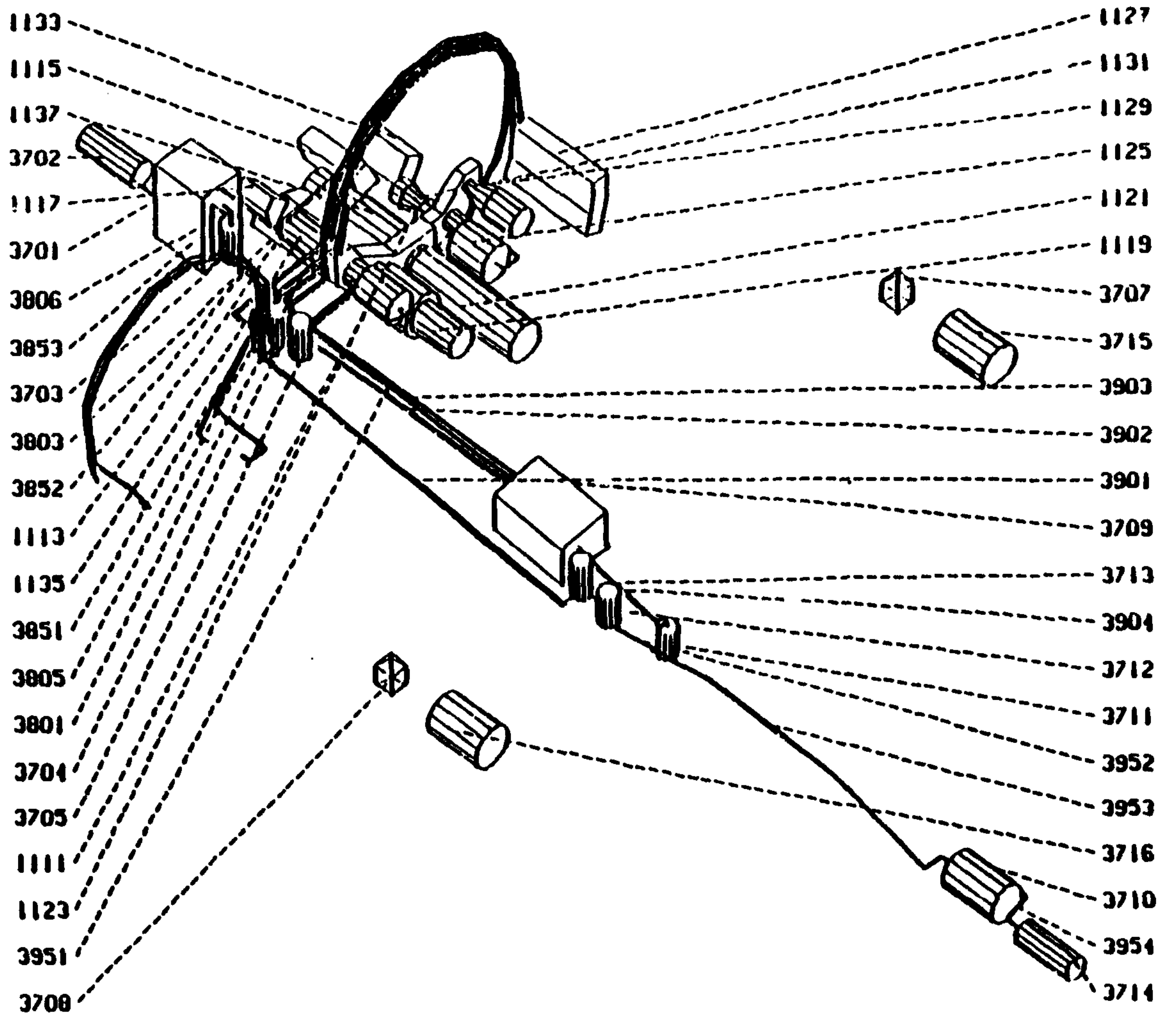


Figure 61. PIXPL Plot Showing Component Code Numbers of Selected Engine and Hydraulic Components

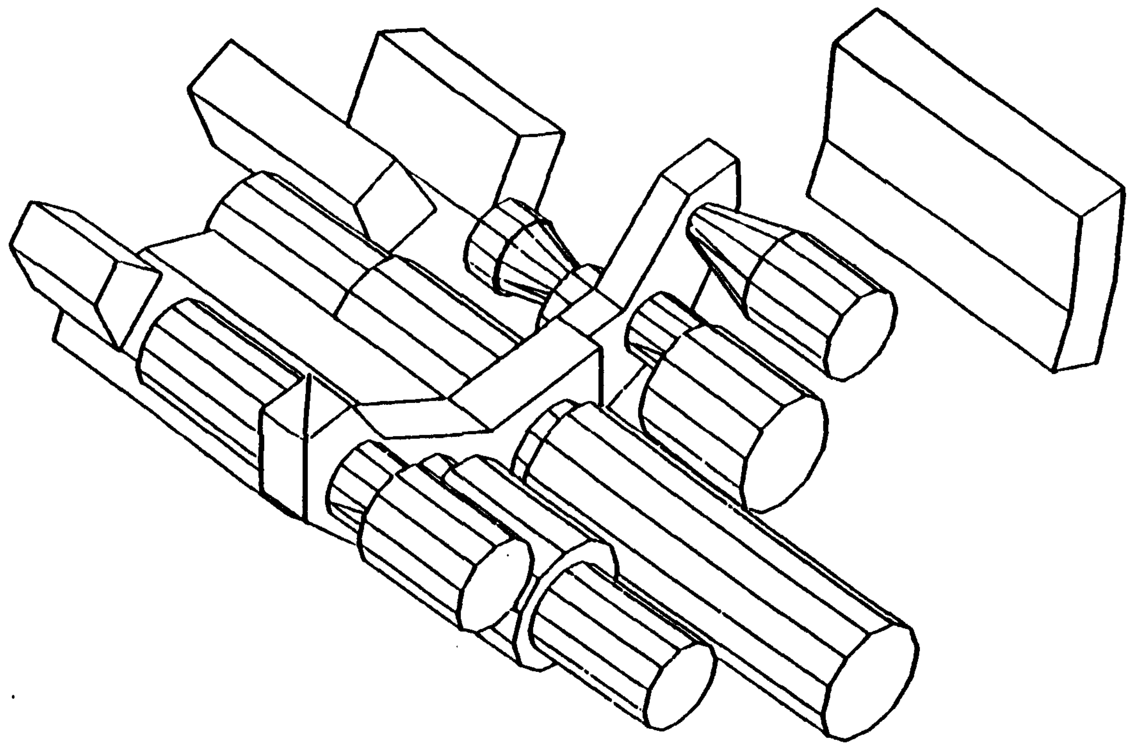


Figure 62. PIXPL Plot of Engine Components Including Gearbox, Afterburner Control Nozzles, A/C Generators, and Hydraulic Pumps

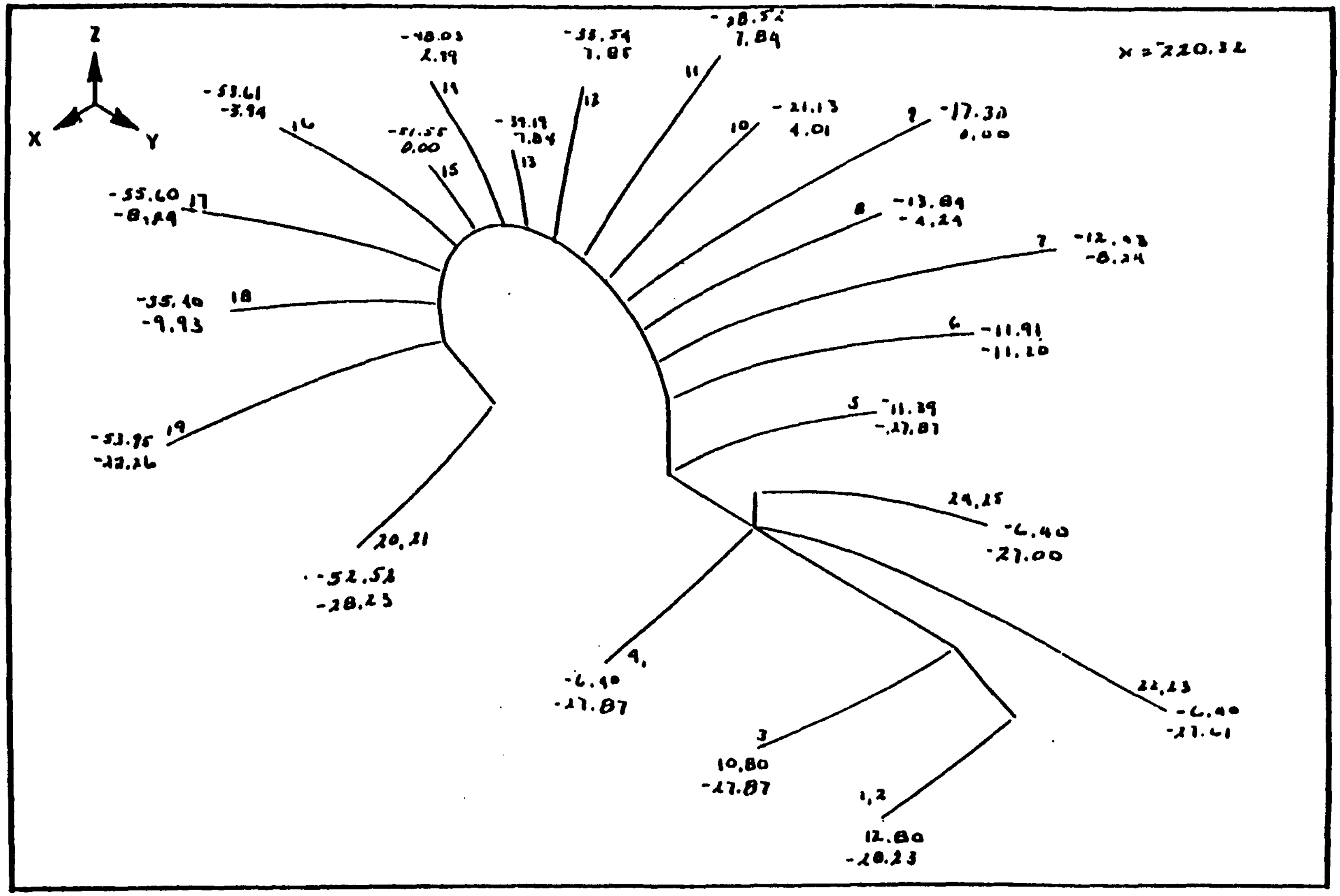


Figure 63. Sketch of Hydraulic Main Pressure Line (Component Code 3851) Described in Rod Mode

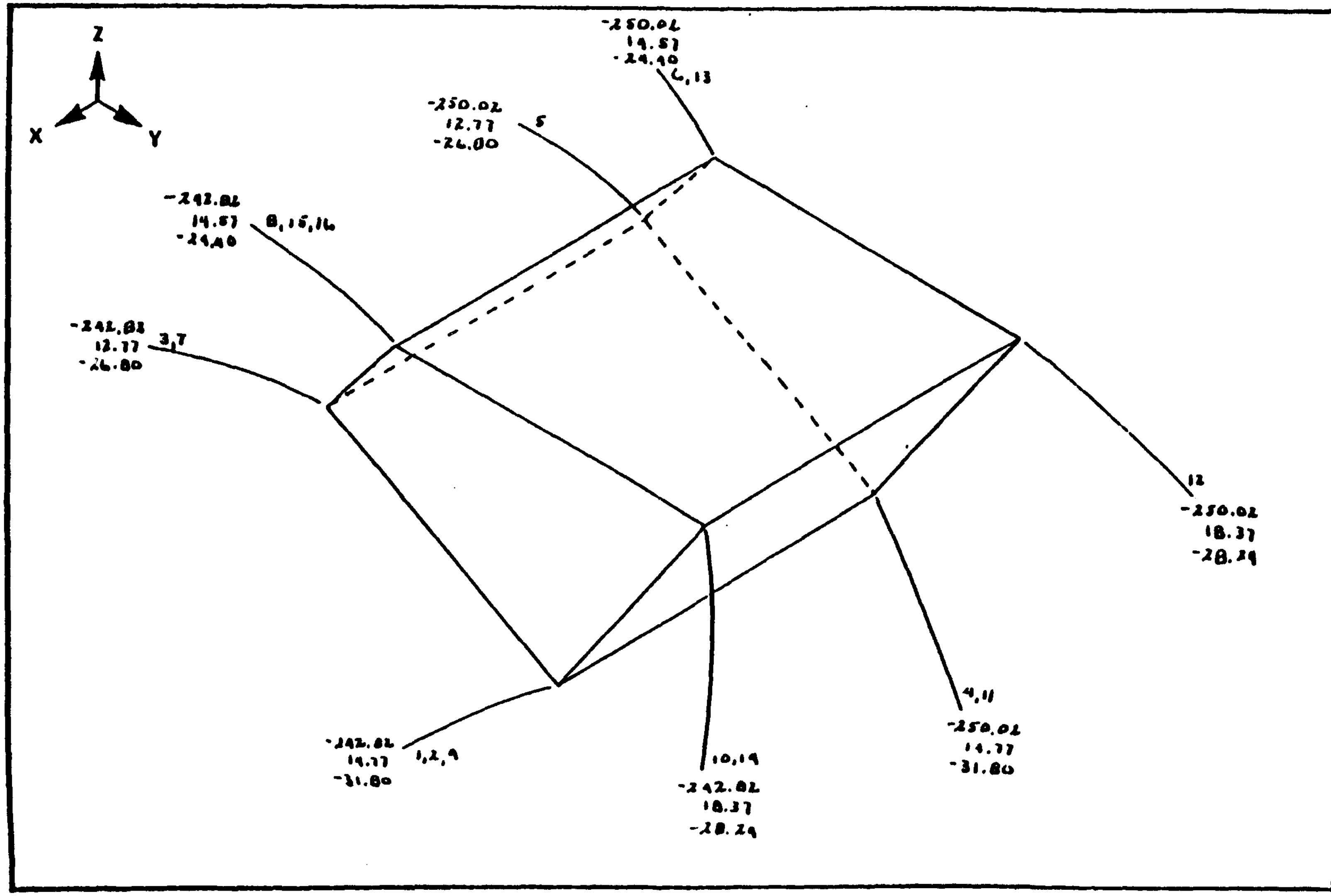


Figure 64. Sketch of Inboard Afterburner Control Box (Component Code 1117) Described with Triangles in Plate Mode

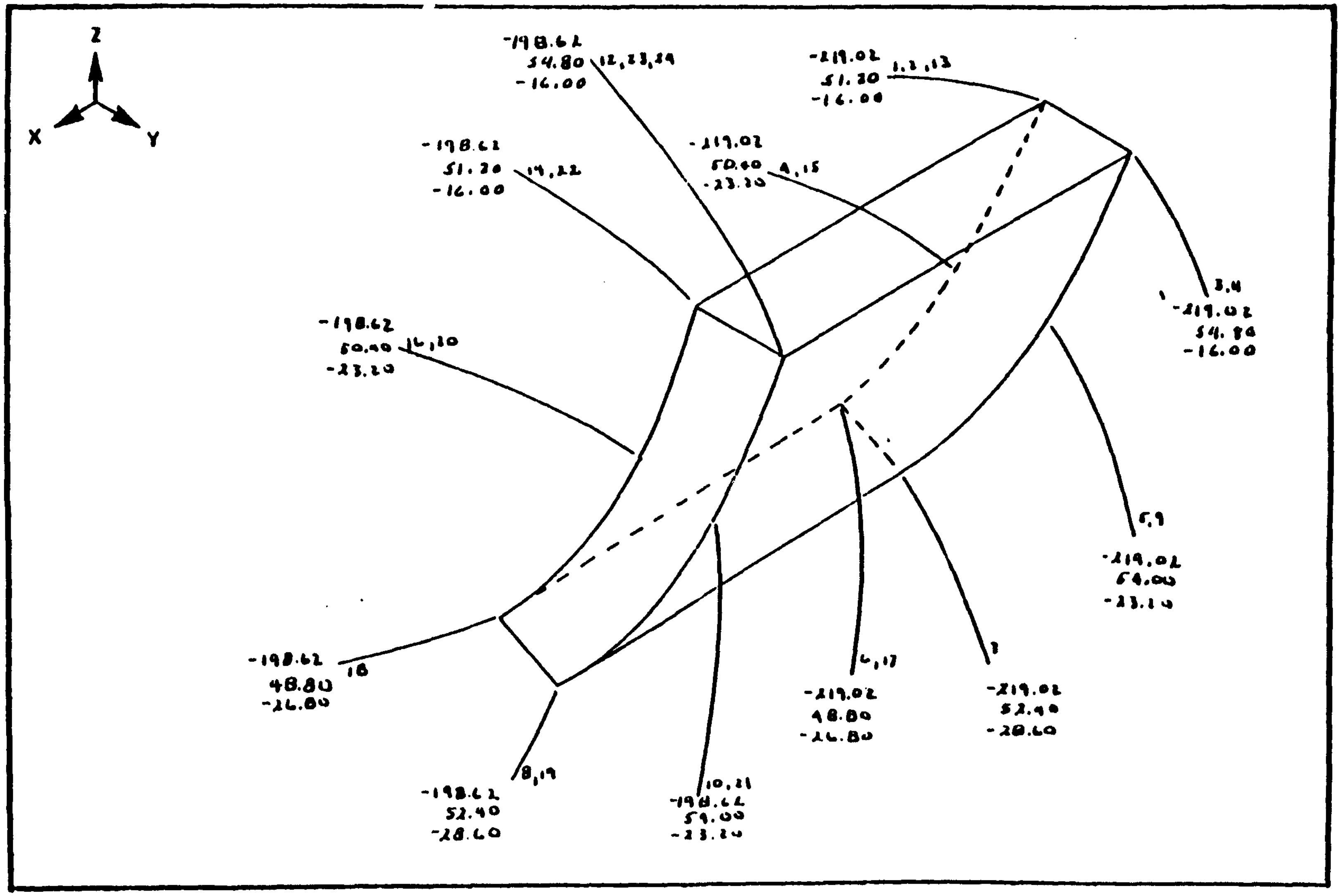


Figure 65. Sketch of Engine Oil Cooler Tank (Component Code 1127) Described with Triangles in Volume Mode

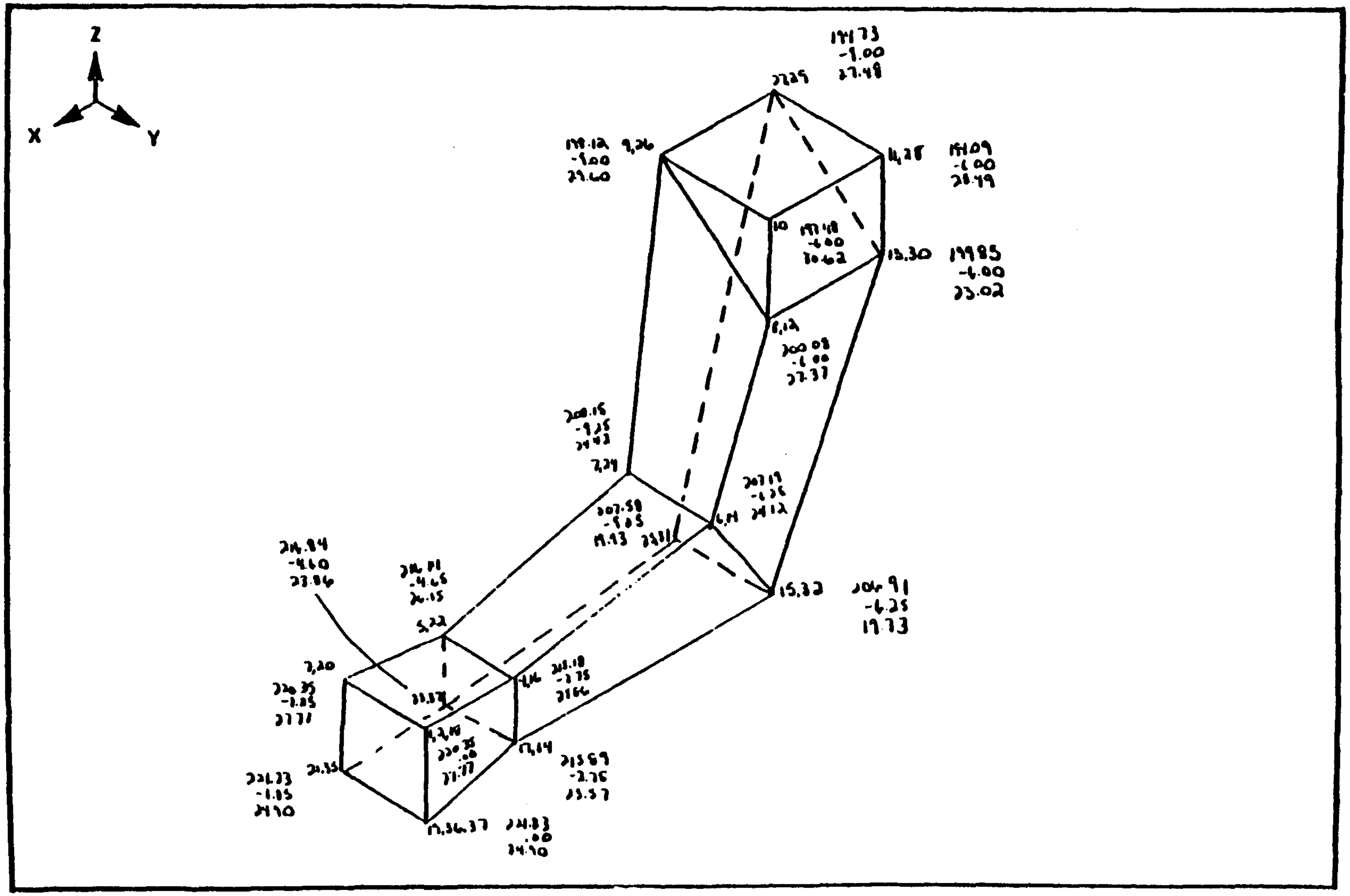


Figure 66. Sketch of Pilot's Right Arm (Component Code 2006) Described with Triangles in Volume Mode

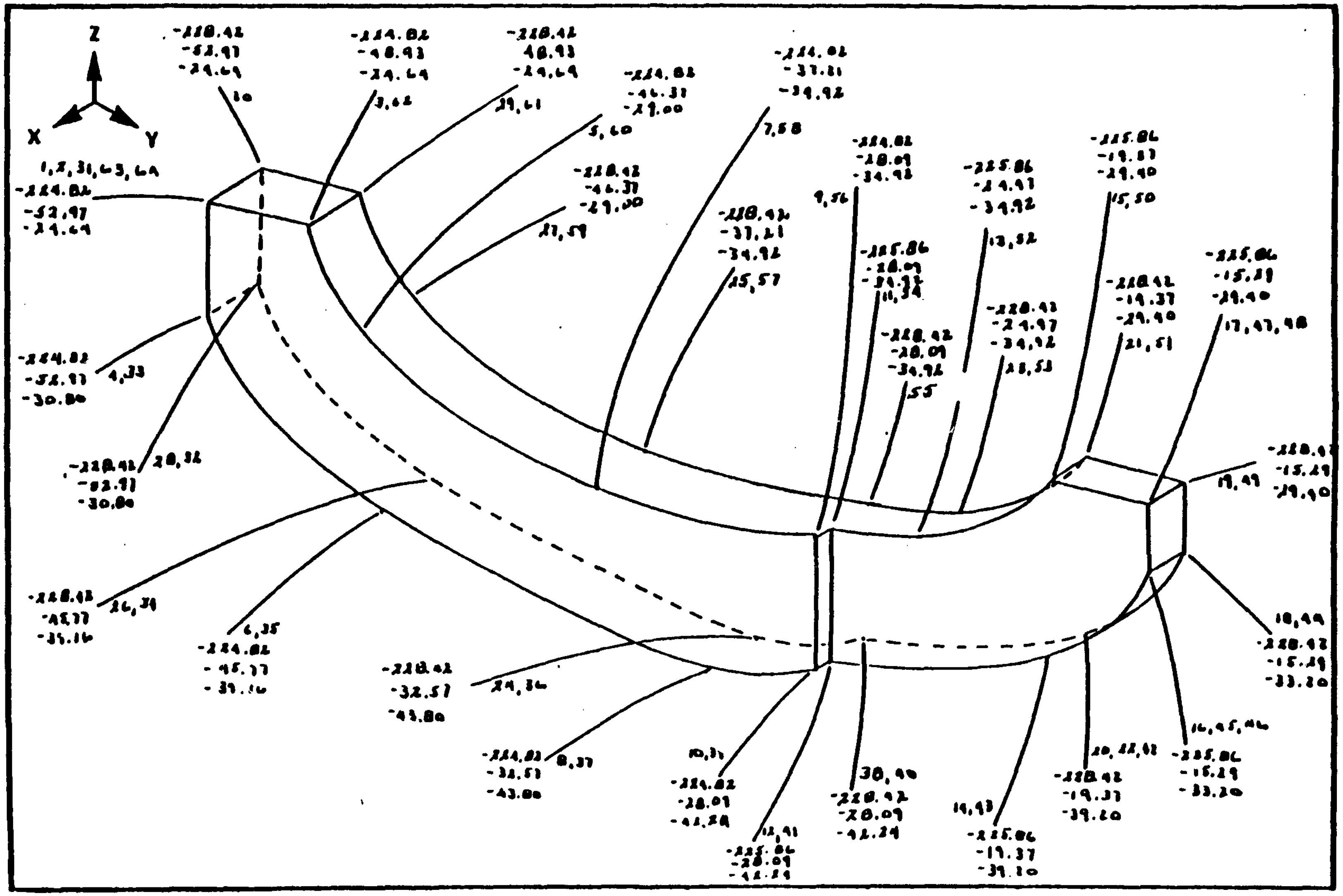


Figure 67. Sketch of Right Engine Gearbox (Component Code 1112) Described with Triangles in Plate Mode

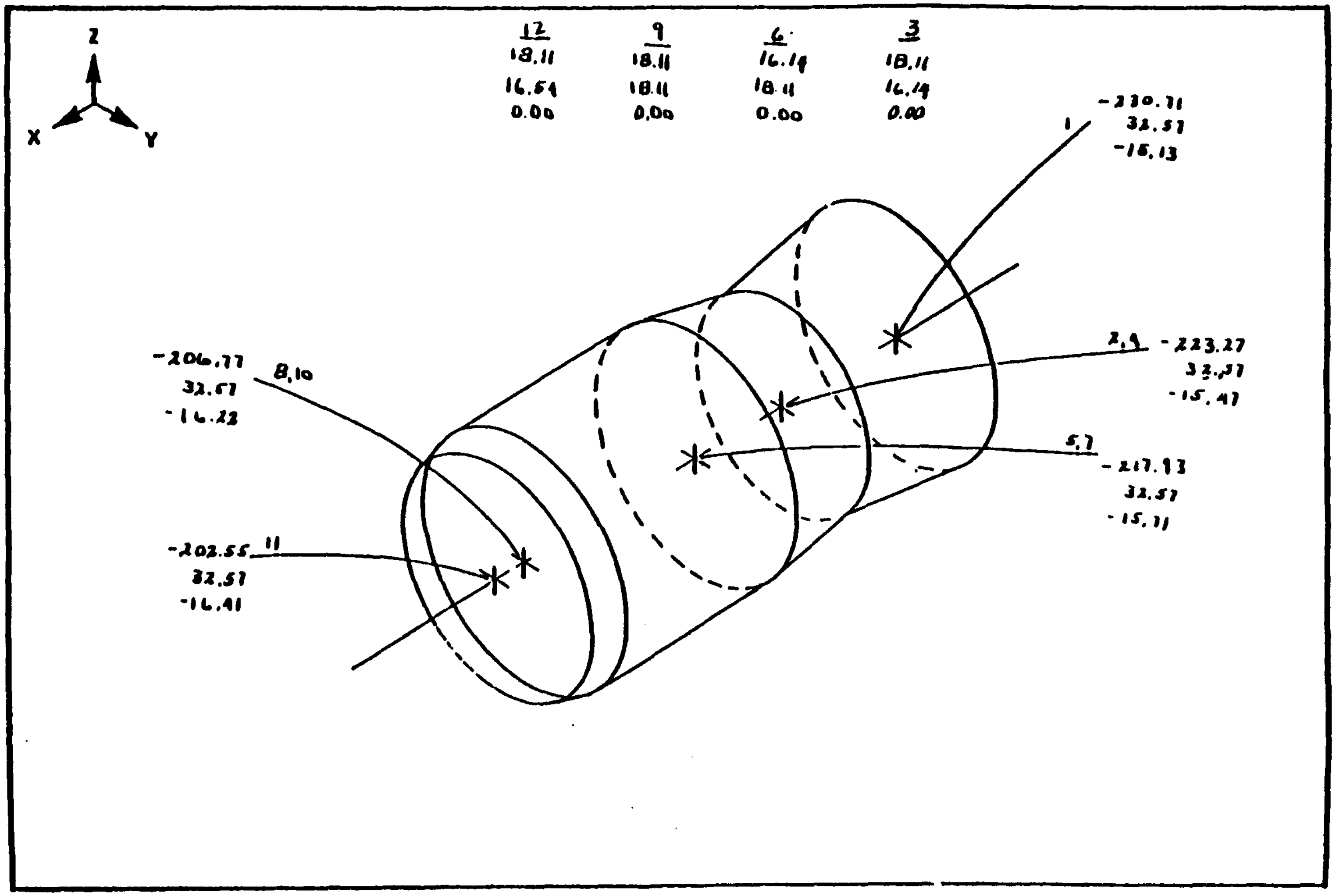


Figure 68. Sketch of Left Engine Compressor Casing (Component Code 1011) Described with Truncated Cones and Cylinders in Plate Mode and Mirrored to Right Engine with Component Code 1012

SECTION IV

MODEL VALIDATION AND CHECKOUT

1. ATTENDANT PROGRAMS

Target modeling can be characterized as a continuous validation process. Exterior surfaces are normally described first to establish the correct size and shape of the target. The TGM matures as components are embedded to replicate the functional aspects of the target. Conflict between components and errors of omission/commission are a perpetual problem for the modeler. However, there are a number of useful tools which can be used to "debug" the TGM. Referring back to Figure 4 in Section 1 CONVERT, PIXPL, FASTGEN, and PLOT5 are indispensable programs and form an integral part of model development, validation and checkout. CONVERT is exercised for general diagnostic tests of the TGM coding. Once a clean CONVERT run is established the other three programs are used to check for component design, placement, and overall modeling constructs.

2. TARGET DEBUGGING

FASTGEN reports four different error codes and suggested methods for correction that can occur along the shotline (Figure 68). It also reports components that have less than one triangle, zero cone height components, and rods with less than one segment. These errors typically occur from typographical errors or miscoding. Usually a quick perusal of these components in the target description deck will show the error. The shotline error codes may take a little more investigation; but, always refer back to the raw data for the components being flagged by FASTGEN.

At first, to debug the TGM, it is desirable to execute FASTGEN using one of the three cardinal views (zero-degree azimuth and elevation; 90-degree azimuth, zero-degree elevation; and 90-degree azimuth and elevation) to aid in interpreting component intercepts. Use large grid squares to check for flagrant modeling errors, i.e. typographical errors resulting in miss typed component code numbers, missing negative signs, erroneous geometric codes, etc. Two or three wrong components can cause several hundred errors so it is advisable to run one view at a time and correct errors as they are detected. Always make corrections to the raw data and then rerun CONVERT. As the target becomes more error free, PIXPL and PLOT5 may be exercised to check component design and placement. Just because FASTGEN is not

ERROR CODES	SUGGESTED METHOD FOR CORRECTION
1 = Incorrect space code	Check target description input Check for previous missing volume surface or interference Change space code to 0 Change component code to 900-999
2 = Missing volume surface	Check sequencing in target description Check CONVERT output Check for line or edge hit Check volume armor component pairs input Check for previous interference Check cylinder/cone end openings
3 = Unallowable interference	Check intentional interference input Check for previous missing surfaces Check for allowable X, Y, Z, maximum and minimum Check target description input
4 = Ambiguous interference	Check intentional interference input Check for previous unallowable interference Check for previous missing surfaces Check allowable X, Y, Z, maximum and minimum Check target description input

Figure 68. FASTGEN Error Codes and Suggested Methods of Correction

reporting errors, does not necessarily mean the TGM is correct. Components may not be in the location the modeler intended, i.e. missing negative signs, or the component is too large or improperly designed. Picture plots are extremely useful and errors often become readily apparent upon careful analysis of the components when plotted. This process is continued until the target is relatively error free. Then FASTGEN runs should be fined tuned, i.e. off-angle views with one to two inch grids for small targets and off-angle views with larger grids for larger targets, whatever is feasible.

If interferences are being reported, these should be investigated to determine whether they are intentional or not. If satisfied that the interference is intentional then the volume

component and the interfering component(s) should be entered into the target deck under the section "intentional interferences". Do not use this option as a means of sanitizing the target without first being assured that the interference is intentional, i.e. a fuel line inside a fuel tank. If it is unintentional, then the model must be corrected.

Debugging the target should continue until ideally it is free of errors. It is an accepted standard to debug the target until a maximum of two percent of the shotlines contain errors. This error rate is predicated on the analyst using several viewing angles while executing the program in a small (2-inch) grid mode.

The final proof of the target description is in the vulnerability analysis. Although the vulnerability model inputs drive the result, incorrect target modeling may manifest itself during actual production runs. The modeler is advised to perform a series of validation runs with the vulnerability model to be used for the vulnerability analysis before production runs begin. Debugging options available in the various vulnerability programs are often helpful in checking the target models.

INTENTIONALLY LEFT BLANK.

APPENDIX A

MATERIALS CODE LIST

The following is a list of materials and code numbers extracted from COVART 3.0

1	Steel (BHN = 100)	23	Lexon
2	Steel (BHN = 150)	24	Cast plexiglass
3	Steel (BHN = 200)	25	Stretched plexiglass
4	Steel (BHN = 250)	26	Doron
5	Steel (BHN = 300)	27	Bullet resistant glass
6	Steel (BHN = 350)	28	Hard rubber
7	Titanium	29	Soft rubber
8	Aluminum 2024	30	Depleted uranium
9	Aluminum 5083	31	Steel (BHN-550)
10	Aluminum 5154	32	Steel (BHN=600)
11	Aluminum 5356	33	Tungsten
12	Aluminum 6061	34	Phenolic
13	Aluminum 7075	35	Oak
14	Aluminum 7039	36	Pine
15	Magnesiumteel	37	Graphite/epoxy
16	Face hardened	51	Water
17	Cast iron	52	Gasoline
18	Copper	53	Lubrications
19	Lead	54	JP1 fuel
20	Tuballoy	55	JP4 fuel
21	Unbonded nylon	56	JP5 fuel
22	Bonded nylon		

INTENTIONALLY LEFT BLANK.

APPENDIX B

TARGET GEOMETRIC MODEL INPUT DECK

The following is a description of the different sections of a sample TGM deck.

$VEHICLE Target identifier.

$COMMENT These cards will contain general target description information i.e., target creation date, latest modification date, target configuration, any target peculiarities, etc.

$NARM If no data exist for components greater than 2.99 in the normal thickness field, the $NARM cards may be omitted. If $NARM cards are present, two cards must follow the $NARM card (16I5, maximum of 32).

$INTERFE Intentional interference cards. If no data exist these cards may be omitted. If data exist the deck must be followed with a $END card. The format is 16I5 where the first field contains the volume component code number; the second field contains the number of interfering components; the third through 16th fields contain the component code numbers of the interfering components.

$TARGET Flags the beginning the raw target data. Formats for raw data are contained in Figure 8, Section II of this report. The last card in the target deck must contain a zero in the ICO field.

$CODE Flags the beginning of the component code list which includes the material codes from Appendix A and the density for each component.

Only the $NARM, $INTERFE, $TARGET decks are to be passed to FASTGEN.

Sample target input deck.

```
$VEHICLE Target identification
$COMMENT There is no limit on the number of $COMMENT cards
$COMMENT Creation date, modification date
$COMMENT Gear down, parked aircraft
$COMMENT
$NARM
 0301 0302 0303 0304
```

$INTERFER
3001 3 9901 9902 9903
3002 5 9904 9905 9906 9907 9908
3003 2 9901 9902
$END
$TARGET

10.0	10.0	10.0	70010001	10	0	0	0	0	0	0	0	0	4
20.0	10.0	10.0	70010001	20	0	0	0	0	0	0	0	0	4
10.0	20.0	10.0	70010001	30	0	0	0	0	0	0	0	0	4
10.0	10.0	20.0	70010001	40	0	0	0	0	0	0	0	0	4
.	.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.
-10.0	-10.0	10.0	80019999	10	0	12	0	0	0	0	0	0	4
-10.0	-10.0	20.0	80019999	20	0	12	0	0	0	0	0	0	4
2.0	2.0	0.0	80019999	30	0	12	0	0	0	0	0	0	4
0.0	0.0	0.0	0	0	0	0	0	0	0	0	0	0	0

$CODE

COMPONENT CODE

0XXX	SKIN
1XXX	ENGINE
2XXX	CREW
3XXX	FLIGHT CONTROLS
4XXX	FUEL SYSTEMS
5XXX	AMMUNITION
6XXX	ARMAMENT
7XXX	STRUCTURE
8XXX	ELECTRONICS AND AVIONICS
9XXX	MISCELLANEOUS

Component Number/Description		Material	Density
0XXX	SKIN		
	Fuselage		
0001	Radome	26	100
0002	Sta -68.90 to Sta -113.78	8	100
0003	Sta -113.78 to Sta -148.82	8	100
0004	Sta -148.82 to Sta -225.59	8	100
0005	Sta -225.59 to Sta -264.57	8	100
0006	Sta -264.57 to Sta -299.21	8	100
0007	Sta -299.21 to Sta -315.75	8	100
0008	Sta -315.75 to Sta -366.14	8	100
0009	Sta -366.14 to Sta -401.18	8	100
0010	Sta -401.18 to Sta -454.72	8	100

Component Number/Description		Material	Density
0011	Sta -454.72 to Sta -512.99	8	100
0012	Engine Exhaust Pipe, Left	7	100
0013	Engine Exhaust Pipe, Right	7	100
0014	Aft End, Center	8	100
.	.	.	.
.	.	.	.
.	.	.	.
	Phantom Armor		
0901	Air Intake, Left	8	0
0902	Air Intake, Right	8	0
0903	Engine Exhaust Pipe, Left	8	0
0904	Engine Exhaust Pipe, Right	8	0
0905	Blast Shield Gun Port	8	0
1XXX	ENGINES		
	Fan		
1001	Fan Inlet Case	7	100
1002	Fan Forward Case	7	100
1003	Fan Aft Case	7	100
1010	Inlet Struts	7	50
1011	Inlet Struts	7	50
1012	Inlet Struts	7	50
1013	Inlet Struts	7	50
1020	1st Stage Rotor Blades	7	50
1021	1st Stage Rotor Disk	7	100
1022	1st Stage Rotor Stator	7	50
.	.	.	.
.	.	.	.
.	.	.	.
2XXX	PILOT		
2001	Head and Neck	51	100
2002	Thorax	51	100
2003	Abdomen	51	100
2004	Pelvis	51	100
2005	Left Arm and Hand	51	100
2006	Right Arm and Hand	51	100
2007	Left Leg and Foot	51	100
2008	Right Leg and Foot	51	100
3XXX	FLIGHT CONTROLS AND HYDRAULICS		
30XX	Main Control Linkages		
3001	Control Stick	8	20
3002	Control Stick Base Yoke	4	50

Component Number/Description		Material	Density
3003	Control Stick Yoke Support	4	50
3004	Left Rudder Pedal	12	100
.	.	.	.
.	.	.	.
.	.	.	.
4XXX	FUEL		
41XX	Fuel Tanks		
4101	Left Wing Fuel	55	1
4102	Right Wing Fuel	55	1
4110	Fwd Fuselage Fuel (0 - 61+%)	55	100
4111	Fwd Fuselage Fuel (61+ - 100%)	55	1
4112	Fwd Fuselage Fuel Tank Top	8	100
4113	Fwd Fuselage Fuel Tank Bottom	13	100
.	.	.	.
.	.	.	.
.	.	.	.
6XXX	ARMAMENT		
60XX	Suspension Equipment		
6001	Left Outboard Missile Rail/Pylon	2	100
6002	Right Outboard Missile Rail/Pylon	2	100
6003	Left Midwing Missile Rail/Pylon	2	100
6004	Right Midwing Missile Rail/Pylon	2	100
6005	Left Inboard Missile Rail/Pylon	2	100
6006	Right Inboard Missile Rail/Pylon	2	100
61XX	Missiles		
	Left Outboard Missile		
6101	Seeker	8	25
6102	Guidance	8	12
6103	Warhead	3	40
6104	Motor	8	40
6105	Canards	8	100
6106	Fins	8	100
.	.	.	.
.	.	.	.
.	.	.	.
7XXX	STRUCTURE		
70XX	Fuselage		
	Frames		
7001	Frame -85.90 (bulkhead behind radome)	13	100
7002	Frame -143.23	13	100
7003	Frame -195.79 (canted bulkhead)	13	100
7004	Frame -242.91 (fuselage)	13	100
.	.	.	.
.	.	.	.
.	.	.	.

Component Number/Description		Material	Density
8XXX	AVIONICS		
	IR Targeting		
8001	Dome	24	100
8002	Base	8	50
8003	Altitude Pitot Tube	2	100
8004	Antenna	8	75
8005	Radar Altitude Antenna	22	75
8006	Vertical Fin Avionics Housing, Left	8	100
8007	Vertical Fin Avionics Housing, Right	8	100
.	.	.	.
.	.	.	.
.	.	.	.
9XXX	MISCELLANEOUS		
90XX	Non-primary Skin		
	Wing Gloves		
9001	Sta 95.70 to Sta -113.78, Left	8	100
9002	Sta -95.70 to Sta -113.78, Right	8	100
9003	Sta -113.78 to Sta -148.82, Left	8	100
9004	Sta -113.78 to Sta -148.82, Right	8	100
.	.	.	.
.	.	.	.
.	.	.	.
92XX	Pneumatics		
9201	Pneumatic Box	4	50
9202	Pneumatic Box	4	50
9203	Pneumatic Valve	4	70
9204	Pneumatic Valve	4	70
93XX	Landing Gear		
	Nose Landing Gear		
9300	NLG Strut Housing	6	100
9301	NLG Strut	6	100
9302	Left NLG Attachment Pivot	6	100
9303	Right NLG Attachment Pivot	6	100
9304	Left NLG Tire	28	30
.	.	.	.
.	.	.	.
.	.	.	.
95XX	Crew Station		
9501	Pilot Seat Frame	8	100
9502	Seat Cushion	29	100
9503	Seat Back	8	50
.	.	.	.
.	.	.	.
.	.	.	.

Component Number/Description		Material	Density
96XX	Variable Engine Air Intakes		
9601	Air Intake Door, Left	8	100
9602	Air Intake Door, Right	8	100
9603	Air Intake Door Forward Support, Left	13	100
9604	Air Intake Door Forward Support, Right	13	100
.	.	.	.
.	.	.	.
.	.	.	.

REFERENCES

1. *CONVERT (Program 7054) User Manual*, Air Force Armament Laboratory, Eglin Air Force Base, Florida, July 1978. .

2. John P. Thompson, and Louis E. Smith, *Target Description for Survivability/Vulnerability Assessment, Volume I Geometric Modeling for FASTGEN*, "NWC Technical Memorandum 3487, Volume I", Falcon Research and Development Company for Systems Development Department, Naval Weapons Center, China Lake, California, May 1978. .

3. R. E. Crammer, and Roy Hilbrand, *FASTGEN 3 Target Description Computer Program*, ASD-TR-85-5019, Air Force Systems Command, Wright Patterson Air Force Base, Ohio, 45433-6503, 10 December 1985.

4. *Shot Generator Computer Program, Volume I User Manual*, and *Volume II Analyst Manual*, JTCG/ME-71-5-1 and JTCG/ME-71-5-2, Joint Technical Coordinating Group for Munitions Effectiveness, July 1970.

5. John P. Thompson and Louis E. Smith, *Target Description for Survivability/ Vulnerability Assessment, Volume III Basic Features of the RAWGEN Computer Program*, "NWC Technical Memorandum 3487, Volume III", Falcon Research and Development Company for Systems Development Department, Naval Weapons Center, China Lake, California, May 1978. .

6. H. E. Page, S. R. Click, R. F. Grove, *Point Burst Ray Generator (PGEN) Computer Program*, AFATL-TR-82-64, Datatec, Inc., for the Air Force Armament Laboratory, Eglin AFB, FL, September 1982. .

7. Leonard E. Bruenning, Jr., and Robert F. Gove, *Point Burst Damage Assessment Model (PDAM), Volume 1, User Manual (U)*, Datatec for Air Force Armament Laboratory, Eglin AFB, Florida, 1983.

8. *COVART 3.0 - A Simulation Program for Computation of Vulnerable Areas and Repair Time - User Manual (U)*, Joint Technical Coordination Group for Munitions Effectiveness, 61JTCG/ME-91-31 March 1989.

9. Leonard E. Bruenning, Jr., Alyndia S. Womble, *High-Explosive Incendiary Vulnerability Assessment Model (HEIVAM)*, AFATL-TR-86-38, Datatec, Inc., for the Air Force Armament Laboratory, Eglin AFB, FL, September 1986.

10. Gary A. Brown, *PIXPL Target Plotting Program (P7057)*, AFATL-TR-79-45, Datatec, Inc., for the Air Force Armament Laboratory, Eglin AFB, FL, April 1979. .

11. Unpublished program resident in the Survivability, Vulnerability, Information Analysis Center (SURVIAC) Aeronautical Systems Center, Computer Center, Wright-Patterson AFB, Ohio, 45433.

REFERENCES (CONCLUDED)

12. *High-Explosive Vulnerability Area and Repair Time (HEVART)*, KTR-109-87, KETRON Division of the Bionetics Corporation, April 1987, UNPUBLISHED

13. Vehicle Damage Computer Program, Volumes I, 61 JTCG/ME-78-1, Joint Technical Coordinating Group for Munitions Effectiveness, 1 May 1977.

No. of Copies	Organization
2	Administrator Defense Technical Info Center ATTN: DTIC-DDA Cameron Station Alexandria, VA 22304-6145
1	Commander U.S. Army Materiel Command ATTN: AMCAM 5001 Eisenhower Ave. Alexandria, VA 22333-0001
1	Director U.S. Army Research Laboratory ATTN: AMSRL-OP-CI-AD, Tech Publishing 2800 Powder Mill Rd. Adelphi, MD 20783-1145
1	Director U.S. Army Research Laboratory ATTN: AMSRL-OP-CI-AD, Records Management 2800 Powder Mill Rd. Adelphi, MD 20783-1145
2	Commander U.S. Army Armament Research, Development, and Engineering Center ATTN: SMCAR-IMI-I Picatinny Arsenal, NJ 07806-5000
2	Commander U.S. Army Armament Research, Development, and Engineering Center ATTN: SMCAR-TDC Picatinny Arsenal, NJ 07806-5000
1	Director Benet Weapons Laboratory U.S. Army Armament Research, Development, and Engineering Center ATTN: SMCAR-CCB-TL Watervliet, NY 12189-4050
1	Director U.S. Army Advanced Systems Research and Analysis Office (ATCOM) ATTN: AMSAT-R-NR, M/S 219-1 Ames Research Center Moffett Field, CA 94035-1000

No. of Copies	Organization
1	Commander U.S. Army Missile Command ATTN: AMSMI-RD-CS-R (DOC) Redstone Arsenal, AL 35898-5010
1	Commander U.S. Army Tank-Automotive Command ATTN: AMSTA-JSK (Armor Eng. Br.) Warren, MI 48397-5000
1	Director U.S. Army TRADOC Analysis Command ATTN: ATRC-WSR White Sands Missile Range, NM 88002-5502
(Class. only) 1	Commandant U.S. Army Infantry School ATTN: ATSH-CD (Security Mgr.) Fort Benning, GA 31905-5660
(Unclass. only) 1	Commandant U.S. Army Infantry School ATTN: ATSH-WCB-O Fort Benning, GA 31905-5000
1	WL/MNOI Eglin AFB, FL 32542-5000
	Aberdeen Proving Ground
2	Dir, USAMSAA ATTN: AMXSY-D AMXSY-MP, H. Cohen
1	Cdr, USATECOM ATTN: AMSTE-TC
1	Dir, ERDEC ATTN: SCBRD-RT
1	Cdr, CBDA ATTN: AMSCB-CII
1	Dir, USARL ATTN: AMSRL-SL-I
5	Dir, USARL ATTN: AMSRL-OP-CI-B (Tech Lib)

No. of
Copies Organization

3 Commander
Naval Air Systems Command
Code AIR-516J (Mr. John Over)(3 cp)
JTCG/AS Central Office
1421 Jefferson Davis Highway
Arlington, VA 22234-5160

1 Commander
Dahlgren Division
Naval Surface Warfare Center
ATTN: Code G29, Mr. Thomas Wasmund
Dahlgren, VA 22448-5000

1 ASC/XREWA (Dr. Kevin T. McArdle)
Eglin AFB, FL 32542-5600

Aberdeen Proving Ground

4 Dir, USAMSAA
ATTN: AMXSY-AD, Charles N. Alston (3 cp)
AMXSY-J, Arthur LaGrange

10 Dir, USARL
ATTN: AMSRL-SL-B, P. Deitz
AMSRL-SL-I, D. Haskell
AMSRL-SL-BA,
A. Kiwan
J. Walbert
S. Polyak
D. Ten Broeck
W. Thompson
R. Walther
J. Fries
K. Kim